Interactive Text Ranking with Bayesian Optimisation: A Case Study on Community QA and Summarisation

Anonymous EMNLP-IJCNLP submission

Abstract

For many NLP applications, such as question answering and summarisation, 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-specific training data, we propose an interactive text ranking approach that actively selects pairs of candidates for a user to compare and label the best. Unlike previous interactive learning strategies, which attempt to learn a ranking across the whole candidate space, our method employs Bayesian optimisation to focus the user's labelling effort on high quality candidates. By integrating prior knowledge in a Bayesian manner, our method is robust to small data scenarios where alternative methods fail. We apply the method to community question answering (cQA) and extractive summarisation, finding that our method significantly outperforms existing interactive text ranking 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 summarisation.

1 Introduction

Many text ranking tasks are highly subjective or context-dependent, yet information about the user's needs is often not readily available to the NLP system. Consider ranking summaries or answers to non-factoid questions in order to find the best candidates: the answer the user wants depends on what they already know and their current information needs (Liu and Agichtein, 2008; López et al., 1999). In this paper, we address this problem by proposing an interactive text ranking approach that efficiently gathers user feedback and combines it with predictions from pre-trained, generic models.

To minimise the amount of effort the user must expend in training a ranker, we learn from pairwise labels, in which the user compares two candidates and labels the best one. Pairwise labels can often be provided faster than ratings or categorical labels (Yang and Chen, 2011; Kingsley and Brown, 2010; Kendall, 1948), and can then be used to rank candidate texts 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 uses one of many different strategies to choose the data points to be labelled, given the set of labels collected so far (Settles, 2012). While active learning aims to to learn a good ranking of all candidates, we are often interested only in sorting the *good* candidates. For instance, in question answering, irrelevant answers may be filtered out and not shown to the user, so their order is unimportant. Therefore, while active learning may reduce labelling costs, some labels may be still wasted on sorting poor candidates.

Here, we propose an interactive text ranking method using *Bayesian optimisation (BO)* (Močkus, 1975; Brochu et al., 2010) instead of AL, which tries to learn the best candidate from a minimal number of labels, rather than trying to learn the entire ranking function. Like AL, BO queries an oracle and trains a model iteratively. At each step it uses the model to estimate the value of the current best candidate, then samples candidates chosen by an *acquisition function* to have a high probability of improving on the current best value.

While BO requires a Bayesian model to perform optimisation, AL can also be used to learn a Bayesian model. A key advantage for interactive tasks is that Bayesian methods can integrate prior knowledge such as heuristics or 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 only after active learning is complete (Gao et al., 2018), or require expensive re-training of a non-Bayesian method, which precludes Bayesian optimisation (Peris and Casacuberta, 2018). In contrast to these approaches, we show how a Bayesian model for text ranking can use prior information to expedite the learning process.

Our contributions are (1) a Bayesian optimisation methodology for interactive text ranking that integrates prior predictions with small amounts of user feedback, (2) acquisition functions for Bayesian optimisation with pairwise labels, and (3) empirical evaluations on community question answering (cQA) and extractive multi-document summarisation, which show that our method brings substantial improvements in ranking and summarisation performance (e.g. for cQA, an average 25% increase in answer selection accuracy over best alternative). We release the complete experimental software to facilitate future research.

2 Background and Related Work

Interactive Learning in NLP. In NLP, previous work has applied active learning to tasks involving ranking or optimising text, including summarisation (P.V.S. and Meyer, 2017), visual question answering (Lin and Parikh, 2017), and translation (Peris and Casacuberta, 2018). For summarisation, Gao et al. (2018) proposed using active learning to learn a reward function for reinforcement learning, which reduces the number of user interactions required to train a reinforcement learner from $\mathcal{O}(1e5)$ to $\mathcal{O}(100)$ compared to reinforcement learners querying the user directly, as in (Sokolov et al., 2016; Lawrence and Riezler, 2018; Singh et al., 2019). This previous work all uses *uncertainty sampling* strategies for AL, which query the candidates with the most uncertain labels to learn the ratings for all candidates. We instead propose to find good candidates using Bayesian optimisation. Recently, Siddhant and Lipton (2018) carried out a large empirical study of uncertainty sampling for sentence classification, semantic role labelling and named entity recognition, finding that exploiting the model uncertainty estimates provided by Bayesian neural networks improved performance, which highlights the strength of Bayesian approaches for interactive settings.

Preference Learning. Preference learning infers rankings from pairwise comparisons by assuming a *random utility model* (Thurstone, 1927), where users choose a candidate from a pair with probability p, where p is a function of the candidates' utility. When candidates have similar utilities, the user's choice will be close to random, while pairs of candidates with very different utilities will be labelled consistently. Two popular preference learning models build on the random utility model: 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:

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

where $\phi(a)$ is the feature vector of a and w^T is a weight parameter that must be learned. Since $a \succ b$ is a binary label, standard cross entropy loss can be applied to learn the weights. The resulting linear model can be used to predict labels for any unseen pairs, as well as to estimate the utility of each candidate, $f_a = w^T \phi(a)$, which can be used to rank the candidates.

Chu and Ghahramani (2005) proposed *Gaussian process preference learning (GPPL)*, a Bayesian nonlinear model based on the Thurstone–Mosteller model. Whereas the BT model described about simply estimates the value of f_a for each candidate, GPPL outputs a posterior distribution over the utilities, $\boldsymbol{f}$, of all candidate texts, $\boldsymbol{x}$:

$$p(\boldsymbol{f} | \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{D}) = \mathcal{N}(\hat{\boldsymbol{f}}_{\boldsymbol{D}}, \boldsymbol{C}_{\boldsymbol{D}}), \quad (2)$$

where $\boldsymbol{D}$ is a set of pairwise preference labels, $\hat{\boldsymbol{f}}_{\boldsymbol{D}}$ is the vector of means of the utilities, and $\boldsymbol{C}_{\boldsymbol{D}}$ is the posterior covariance matrix of the utilities. The posterior mean, $\hat{\boldsymbol{f}}_{\boldsymbol{D}}$, contains the predictions of f_a for each candidate, and the covariance, $\boldsymbol{C}_{\boldsymbol{D}}$, represents confidence in the predictions.

Gaussian processes have been shown to reduce errors with sparse or noisy labels (Cohn and Specia, 2013; Beck et al., 2014) making them well suited to learning from user feedback. To enable inference with arbitrarily large numbers of candidates and pairs, Simpson and Gurevych (2018) introduced stochastic variational inference for GPPL and found that it outperformed SVM and LSTM methods at ranking arguments by convincingness. Here, we adapt GPPL to summarisation and cQA, and propose a new Bayesian optimisation framework for GPPL for that enables text ranking in an interactive setting.

Bayesian Optimisation for Preference Learning Brochu et al. (2008) proposed a BO approach for pairwise comparisons but applied the approach only to a simple material design use case. Building on this work, González et al. (2017) proposed alternative strategies for BO using pairwise preferences, however, their approach requires an expensive sampling method to estimate the utilities, which is too slow for an interactive setting, as they do not use GPPL to infer the utilities directly. Recent work by Yang and Klabjan (2018) also proposes Bayesian optimisation with pairwise preferences using a deep Gaussian process model. However, inference is expensive, the method is only tested on data with fewer than ten features, and it uses an inferior Bayesian optimisation strategy (see the comparison of *probability of improvement* and *expected improvement* in Snoek et al. (2012)). In contrast, our GPPL-based framework permits much faster inference that 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 optimiser 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 ranking text, nor applied BO to interactive text ranking tasks.

3 Active Preference Learning

Uncertainty of Optimality (UNC). During active learning a learner iteratively requests training labels for candidates that maximise an *acquisition function* (Settles, 2012). P.V.S. and Meyer (2017) proposed an uncertainty sampling acquisition function for interactive document summarisation, which can be defined as:

$$u_{opt}(a|\boldsymbol{D}) = \begin{cases} p_{opt}(a|\boldsymbol{D}) & \text{if } p_{opt}(a|\boldsymbol{D}) \leq 0.5 \\ 1 - p_{opt}(a|\boldsymbol{D}) & \text{if } p_{opt}(a|\boldsymbol{D}) > 0.5, \end{cases} \quad (3)$$

where $p_{opt}(a|\boldsymbol{D}) = (1 + \exp(-\boldsymbol{w}_{\boldsymbol{D}}^T \boldsymbol{\phi}(a)))^{-1}$ estimates the probability that a is the optimal candidate, and $\boldsymbol{w}_{\boldsymbol{D}}$ is the set of BT model weights learned from the data collected so far, $\boldsymbol{D}$. This acquisition function focusses labelling effort on summaries with $p_{opt}(a|\boldsymbol{D})$ close to 0.5, which includes summaries whose quality is uncertain. To apply this acquisition function to pairwise labels, Gao et al. (2018) select one pair per active learning round consisting of the two summaries with the highest values of u_{opt}. After active learning is complete, they also integrate prior predictions based on heuristics, by taking a weighted sum of the BT predictions and heuristic predictions.

The drawback of UNC is that summaries of intermediate quality will have $p_{opt}(a|\boldsymbol{D})$ close to 0.5, even if they have been labelled many times already, so labelling effort may be wasted on these summaries. To address this, we distinguish two types of uncertainty: *epistemic* uncertainty that can be reduced by acquiring more training data, and *aleatoric* uncertainty that remains given infinite data, due to the inherent unpredictability of the labels (Der Kiureghian and Ditlevsen, 2009). For example, if two items have identical features, it will not be possible to predict the winner of their comparison, even with the best possible model. Interactive learning should therefore ideally target items with high epistemic, rather than aleatoric, uncertainty to avoid wasted effort.

Bayesian Active Preference Learning. Since BT does not quantify epistemic uncertainty, we turn to GPPL to develop a Bayesian approach for interactive learning. As described in Section 2, GPPL quantifies the epistemic uncertainty in its predictions through the covariance matrix, $\boldsymbol{C}_{\boldsymbol{D}}$. Below, we utilise this posterior covariance to define new acquisition functions.

We also exploit GPPL to address the cold start problem in interactive learning, in which the model has little information to select the first few queries to the user. 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 $p(\boldsymbol{f}|\boldsymbol{\phi}(\boldsymbol{x})) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{K})$, where $\boldsymbol{K}$ is a prior covariance matrix. Given this setup, AL and BO can now take the prior predictions into account when choosing pairs of candidates for labelling. We now define Bayesian active learning strategies for GPPL, which we later compare to BO.

Pairwise Uncertainty Sampling (UNPA). Rather than evaluating each candidate individually, as in UNC, we select the pair whose label is most uncertain, since the combination of items in a pair affects how much can be learned from it. For example, if we compare two items with identical features, the learner may not learn anything that generalises to other items.

UNPA selects the pair with label probability $p(y_{a,b} = 1)$ closest to 0.5, where, for GPPL:

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

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

where Φ is the probit likelihood and $\hat{f}_a$ is the posterior mean of the utility for candidate a. This accounts for epistemic uncertainty through $\boldsymbol{C}$, but does not distinguish it from aleatoric uncertainty, so it may still waste labelling effort on pairs of items with similar scores but high confidence.

Expected Information Gain (EIG). Choosing pairs that maximise *information gain*, which quantifies the information a pairwise label provides about our model's parameters, greedily reduces the epistemic uncertainty in the model. Unlike UNPA, this function will avoid pairs that have high aleatoric uncertainty only because they have similar expected utilities. 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, so we compute the reduction in entropy of the label's distribution given $\boldsymbol{f}$. Since we do not know the exact value of $\boldsymbol{f}$, we take the *expected* information gain $\boldsymbol{I}$ with respect to $\boldsymbol{f}$:

$$\mathrm{I}(y_{a,b}, \boldsymbol{f}; \boldsymbol{D}) = \mathrm{H}(y_{a,b}|\boldsymbol{D}) - \mathbb{E}_{\boldsymbol{f}|\boldsymbol{D}}[\mathrm{H}(y_{a,b}|\boldsymbol{f})], \tag{5}$$

where H is Shannon entropy. Unlike the approach of González et al. (2017), this equation can be computed in closed form given the GPPL posterior, so does not need expensive sampling.

4 Bayesian Optimisation for Preferences

Expected Improvement (IMP). The previous acquisition functions for AL are uncertainty-based, and spread labelling effort across all items whose utilities are uncertain. However, for tasks such as summarisation or cQA, the goal is to find the best candidates. Hence it is important to distinguish between candidates that are reasonably good, and those that may be the optimum. To address this challenge using BO, we replace the previous acquisition functions with a new function that estimates the *expected improvement* (Močkus, 1975), of an item, a over our current estimated best solution, b, given current pairwise labels, $\boldsymbol{D}$:

$$\mathrm{Imp}(a, b; \boldsymbol{D}) = vz\Phi(z) + v\mathcal{N}(z; 0, 1), \tag{6}$$

$$z = \frac{\hat{f}_{\boldsymbol{D},a} - \hat{f}_{\boldsymbol{D},b}}{\sqrt{v}}.$$

This equation weights the probability of finding a better solution, $\Phi(z)$, by the amount of improvement, z. It accounts for both how close the utility of a is to b, through z, and the uncertainty in both utilities through the variance, v.

To select pairs of items, we choose the current best item and the item with the greatest expected improvement. This is a greedy strategy as it samples items that are expected to lead to greatest improvements according to the current model. It can be seen as balancing *exploration* of unknown candidates with *exploitation* of promising candidates. In contrast, active learning using uncertainty sampling is pure exploration.

Thompson Sampling with Pairwise Labels (TP). Thompson sampling (Thompson, 1933) is an alternative method for balancing the need to exploit the current model and explore possible alternatives. We select an item using Thompson sampling as follows: first draw a sample of utilities for candidate items from their posterior distribution, $\boldsymbol{f}_{tho} \sim \mathcal{N}(\hat{\boldsymbol{f}}_{\boldsymbol{D}}, \boldsymbol{C}_{\boldsymbol{D}})$, then choose the item y_{best} with the highest score in the sample. Note that this sampling step depends on a Bayesian approach to provide a posterior distribution. To create a pair of items for preference learning, we compute the expected information gain for all pairings of y_{best}, and choose the pair with the maximum. Compared to IMP, this strategy is less greedy as it allows for more learning about uncertain items through both the Thompson sampling step and the information gain step. However, compared to EIG, effort is focussed on items with potentially high scores, due to the first step.

5 Experiments

We perform experiments on three tasks to test our interactive text ranking learning approach: (1) community question answering (cQA), where the goal

cQA Topics	#questions	#accepted answers	#candidate answers
Apple	1,250	1,250	125,000
Cooking	792	792	79,200
Travel	766	766	76,600
Summarisation Datasets	#topics	#model summaries	#docs
DUC 2001	28	84	288
DUC 2002	59	177	567
DUC 2004	50	150	500

Table 1: Dataset statistics for summarisation and cQA.

is to identify the best answer to a given question from a pool of candidate answers; (2) rating document summaries, which aims to learn ratings that reflect a user's preferences over summaries; and (3) generating the optimal extractive 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 tens.

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). For summarisation, we use the DUC datasets[1], which contain model summaries for collections of documents related to a topic. Statistics for the datasets are shown in Table 1. For all datasets, we obtain feature vectors for the candidates by taking bag-of-bigram embeddings (Rioux et al., 2014).

Methods. Using GPPL as our preference learner, we compare the different strategies for acquiring data described in Section 4: our proposed Bayesian optimisation methods, IMP and TP; uncertainty-based Bayesian active learning, UNPA and EIG; and random selection. We also compare GPPL against BT using the UNC active learning strategy and random selection.

By default, we select the initial pair in each interactive learning session at random, since the strategies cannot be properly applied when there is no training data. To prevent a poor initial choice that leads to low performance, we introduce a heuristic that chooses the best candidate according to the prior, plus the candidate whose feature vector has the lowest cosine similarity to it. This exploits prior knowledge and maximises diversity in the initial sample,as shown to be effective by Yin et al. (2017). Where random initial pairs or random sampling are used, we repeated each experiment ten times.

Priors. We test two approaches to integrate prior knowledge, i.e. predictions of the utilities computed before any user interactions have been observed: *prior*, where we set the prior mean of GPPL to the value of the prior predictions, and *sum*, where we use a weighted sum to combine the prior predictions with the posterior preference function values learned using GPPL or BT. Based on preliminary experiments, we weight the prior and posterior predictions equally.

In summarisation, we use the heuristics proposed by Ryang and Abekawa (2012) as a prior. For cQA, we obtain prior predictions using a state-of-the-art method, 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 COALA on the training set splits given by Rücklé et al. (2019), then predict on the test sets, i.e., the datasets in Table 1.

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 $p(a \prec b | g_a, g_b) = (1 + \exp(\frac{g_b - g_a}{t}))^{-1}$, where t is a parameter that controls the noise level. In both datasets, we are provided with 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 often used for evaluating question answering systems (e.g. (Nguyen et al., 2016; Bauer et al., 2018; Indurthi et al., 2018)) and test two values of $t = 1$ and $t = 0.1$. For summarisation, we use $t = 1$. As gold, we combine ROUGE scores using the following formula, which was shown to correlate well with human preferences(P.V.S. and Meyer, 2017):

$$g_a \approx R_{comb} = \frac{\text{ROUGE}_1(a,m)}{0.47} + \frac{\text{ROUGE}_2(a,m)}{0.22} + \frac{\text{ROUGE}_{\text{SU4}}(a,m)}{0.18}. \quad (7)$$

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

5.1 Community Question Answering

Despite strong performance relative to other cQA methods, COALA often fails to choose the best answer. We hypothesise that this can be improved by obtaining a small amount of user feedback for each question, then re-ranking candidate answers based on both the COALA predictions and the user feedback. We compute accuracy as the percentage of gold answers that were correctly selected as best answers. To show how similar the best solutions proposed by each method were to the gold answers, we also compute normalised discounted cumulative gain at one (NDCG@1), using ROUGE-L as relevance. 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 Table 2 show that with only 10 user interactions, most methods are unable to improve accuracy over pre-trained COALA, but do improve NDCG@1 with a low-noise simulated user. This suggests that the user feedback with UNC, UNPA, EIG and TP helps to find answers similar to the accepted answer, but does not allow GPPL to distinguish the top item correctly. However, when using the IMP strategy, both the accuracy and NDCG@1 are significantly higher than the pre-trained values, even when the user provides noisy labels ($p \ll .01$ using a two-tailed Wilcoxon signed-rank test). Including COALA predictions as the GPPL *prior* is highly beneficial for IMP, as the predictions help to identify the best candidates for further sampling, but does not help UNPA, as the priors do not help to estimate the uncertainty in the candidates utilities. Overall, our IMP strategy is extremely effective at leveraging the priors and choosing queries that identify the best answer within a small number of interactions, while other methods fail.

5.2 Interactive Summary Rating

We apply interactive learning to refine a ranking over candidate summaries given prior information. For each topic, we create 10,000 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 both 10 and 100 user interactions with noisy LNO-1 simulated users. On a standard Intel desktop workstation with a quad-core CPU and no GPU, we find that updates to GPPL at each interactive learning iteration require around one second, hence the method is suitable for interactive scenarios.

		Strat	Apple		Cooking		Travel	
Learner	Prior	-egy	acc	N1	acc	N1	acc	N1
COALA	pre-trained		31.8	.419	47.8	.558	52.8	.606
Low noise, #interactions=10, random initial pair								
BT	sum	random	27.2	.416	36.8	.476	41.0	.509
BT	sum	UNC	23.3	.381	30.8	.426	34.7	.455
GPPL	sum	random	24.5	.399	34.1	.458	39.3	.500
GPPL	sum	UNPA	29.3	.443	45.1	.552	42.3	.526
GPPL	sum	IMP	37.3	.524	46.9	.580	46.6	.573
GPPL	*prior*	IMP	55.5	.684	70.8	.780	74.8	.812
Low noise, #interactions=10, heuristic initial pair								
GPPL	pm	random	35.2	.494	48.9	.586	55.6	.640
GPPL	pm	UNPA	29.0	.438	39.2	.504	47.6	.571
GPPL	pm	EIG	30.2	.452	37.2	.488	46.9	.565
GPPL	pm	TP	27.4	.429	35.3	.471	41.4	.518
GPPL	pm	IMP	**61.5**	**.728**	**75.0**	**.813**	**78.4**	**.841**
Noisy, #interactions = 10, random initial pair								
GPPL	pm	UNPA	29.2	.379	39.2	.472	45.7	.530
GPPL	pm	IMP	50.6	.572	61.4	.666	63.2	.687

Table 2: Interactive text ranking for cQA. “N1” is NDCG@1, “acc” is accuracy. The low noise setting simulates an LNO-0.1 user with 83% accuracy (fraction of times the pairwise label corresponded to the gold ranking); noisy uses LNO-1 with 58% accuracy.

We evaluate the quality of the 100 highest-ranked summaries using NDCG@1%. Besides using the inferred utilities to obtain a ranking, the values themselves can be useful for filtering and as rewards for reinforcement learning. We therefore evaluate the predicted utilities by computing the Pearson correlation with the combined ROUGE scores (Equation 7).

The results in Table 3 show a clear advantage to IMP with both 10 and 100 interactions, which outperforms the previous state of the art, BT with UNC (significant with $p \ll .01$ on DUC’01 and DUC’02 with 10 interactions for NDCG@1%, on DUC’02 with 10 interactions for r, and both NDCG@1% and r for all datasets with 100 interactions). UNC and UNPA only improve consistently over random sampling with 100 interactions. EIG and TP achieve gains with both 10 and 100 interactions, likely because EIG considers only epistemic uncertainty, unlike UNPA, and the optimisation approach of TP improves performance further.

With only 10 interactions, GPPL-random outperforms BT in terms of r, but under-performs in terms of NDCG@1%, and integrating prior predictions using the GP *prior* improves r over using a weighted *sum*. However, with 100 interactions, GPPL with *prior* has a substantial advantage over BT or GPPL with *sum*. UNPA, EIG and TP also

		Strat	DUC'01		DUC'02		DUC'04	
Learner	Prior	-egy	N1	r	N1	r	N1	r
none	heuristics only		.556	.217	.573	.278	.597	.322
#interactions=10, random initial pair								
BT	*sum*	random	.560	.259	.590	.275	.613	.307
BT	*sum*	UNC	.563	.257	.593	.272	.620	.313
GPPL	*sum*	random	.558	.271	.585	.283	.606	.322
GPPL	*sum*	UNPA	.555	.251	.583	.268	.609	.308
GPPL	*sum*	IMP	.568	.280	.583	.286	.608	**.326**
GPPL	*prior*	IMP	**.591**	**.292**	.605	.284	.621	.300
#interactions=10, heuristic initial pair								
GPPL	*prior*	random	.558	.288	.588	.293	.606	.323
GPPL	*prior*	UNPA	.559	.273	.580	.278	.607	.309
GPPL	*prior*	EIG	.564	.273	.592	.291	.614	.315
GPPL	*prior*	TP	.572	.272	.589	.299	.612	.325
GPPL	*prior*	IMP	.583	.263	**.612**	**.306**	**.625**	.308
#interactions=100, random initial pair								
BT	*sum*	random	.587	.323	.623	.355	.653	.397
BT	*sum*	UNC	.591	.329	.632	.365	.661	.410
GPPL	*sum*	random	.543	.333	.640	.405	.670	.453
GPPL	*sum*	UNPA	.597	.347	.628	.369	.664	.431
GPPL	*sum*	IMP	.641	.367	.675	.388	.621	.332
GPPL	*prior*	IMP	.686	.374	.711	.402	.733	.425
#interactions=100, heuristic initial pair								
GPPL	*prior*	random	.610	.327	.591	.298	.678	.409
GPPL	*prior*	UNPA	.634	.375	.658	.414	.678	.409
GPPL	*prior*	EIG	.649	.389	.670	.414	.694	.450
GPPL	*prior*	TP	.656	.359	.691	**.463**	.731	.416
GPPL	*prior*	IMP	**.695**	**.406**	**.714**	.404	**.741**	**.478**

Table 3: Interactive Summary Rating. “N1” is normalised cumulative discounted gain at 1% of results and “r” is Pearson’s correlation coefficient.

become increasingly effective, but are still outperformed by IMP. To summarise, the IMP strategy is by far the most effective strategy, and benefits from the GPPL prior, particularly with few interactions.

5.3 RL for Summarisation

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 ranking or rewards assigned to bad candidates are correct, as long as they are distinguished from good candidates, as this will prevent the bad candidates from being chosen. To test the hypothesis, we simulate a *flat-bottomed* reward function for summarisation on the DUC'01 corpus: first, for each topic, we set the reward values for the 10,000 sampled summaries (see Section 5.2) to the gold standard score, R_{comb}, defined in Eq. (7). Then, we normalise the rewards to $[0, 10]$ and set the rewards for a varying 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 1. The results show that 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 labelling effort should be spent on the top items as much as possible.

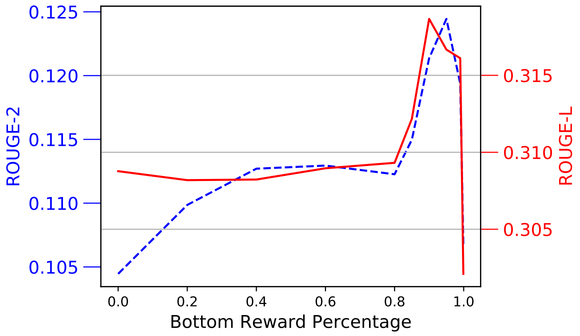

Figure 1: Performance of RL on a summarisation task, when the reward values for the bottom $x\%$ summaries are flattened to one. Dashed line shows ROUGE-2 and solid line shows ROUGE-L.

We now use the ranking functions learned in the previous task as rewards for reinforcement learning. We replicate the RL setup of Gao et al. (2018) for interactive multi-document summarisation, which previously achieved state-of-the-art performance using the BT learner with UNC. The RL agent models the summarisation process as follows: there is a current state, represented by the current draft summary; the agent takes actions depending on the state that modify the draft summary by concatenating new sentences or terminating summary construction. During the learning process, the agent receives a reward after terminating, which it uses to learn a policy (state-to-reward mapping) to maximise 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 the policy's best summary. We then compare the produced summary using ROUGE to a human-generated model summary. By improving the reward function, we hypothesise that the quality of the resulting summary will also improve.

Table 4 shows that the best-performing method from the previous tasks, IMP, again produces a strong improvement over BT with UNC in all cases except for DUC'04 with 10 interactions, where the Pearson correlation was lower. This suggests that while the ranking was been strong on DUC'04, the predicted values of the utilities may have been less

			DUC'01				DUC'02				DUC'04			
Learner	Prior	Strategy	R1	R2	RL	RSU4	R1	R2	RL	RSU4	R1	R2	RL	RSU4
None	heuristics only		.324	.069	.256	.099	.350	.081	.276	.113	.372	.086	.292	.122
#interactions=10, random initial pair														
BT	*sum*	UNC	.337	.072	.266	.104	.362	.085	.286	.119	.388	**.102**	.307	**.134**
#interactions=10, heuristic initial pair														
GPPL	*prior*	random	.325	.068	.256	.097	.350	.083	.277	.110	.380	.088	.302	.127
GPPL	*prior*	UNPA	.344	.082	.273	.108	.351	.084	.280	.113	.384	.089	.303	.128
GPPL	*prior*	EIG	.344	.083	.275	.109	.356	.086	.283	.112	.385	.093	.305	.130
GPPL	*prior*	TP	.344	.081	.271	.108	.352	.084	.275	.111	**.394**	.095	**.310**	**.134**
GPPL	*prior*	IMP	**.349**	**.089**	**.276**	**.111**	**.365**	**.095**	**.288**	**.119**	.390	.095	.308	.133
#interactions=100, random initial pair														
BT	*sum*	UNC	.347	.080	.274	.109	.369	.089	.286	.123	.391	.101	.308	.136
#interactions=100, heuristic initial pair														
GPPL	*prior*	random	.328	.069	.258	.101	.355	.091	.286	.115	.371	.086	.293	.123
GPPL	*prior*	UNPA	.345	.081	.276	.111	.371	.092	.292	.122	.388	.101	.305	.134
GPPL	*prior*	EIG	.351	.084	.275	.114	.378	.098	.299	.126	.398	.110	.317	.141
GPPL	*prior*	TP	.341	.082	.272	.110	.376	.094	.295	.127	.401	.112	.316	.142
GPPL	*prior*	IMP	**.352**	**.098**	**.283**	**.117**	**.394**	**.107**	**.310**	**.136**	**.411**	**.121**	**.325**	**.149**

Table 4: Reinforcement learning for summarisation: ROUGE scores of the chosen summaries.

accurate. EIG and TP also appear to consistently outperform BT with UNC for 100 interactions. The results show that our proposed IMP strategy improves (significant with $p \ll 0.01$ on all cases with 100 interactions and DUC'01 with 10 interactions) on the previous state of the art, and establishes that gains made by Bayesian optimisation when learning the utilities translate to the final summaries produced by reinforcement learning.

6 Discussion and Conclusions

We proposed a novel interactive text ranking setup, which uses Bayesian optimisation to acquire pairwise feedback from a user and train a model using Gaussian process preference learning (GPPL). Our experiments showed that our approach 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 summaries when used to learn a reward function for reinforcement learning.

We showed that a Bayesian optimisation strategy based on expected improvement (IMP) outperforms Thompson sampling. Since Thompson sampling involves a random sampling step, this strategy may be more effective with a larger number of interactions. We found that active learning using uncertainty sampling, which attempts to learn the whole reward function, is less effective than BO for ranking answers or summaries. Note that IMP is effective in both tasks, but has the strongest impact on cQA with only 10 interactions, where other methods produce only modest increases in ranking performance with NDCG@1. This may be due to the greater sparsity of candidates in cQA (100 versus 10,000), which allows them to be more easily distinguished by the model, provided it selects the right training examples.

The Bayesian optimisation strategies depend on a learner that can quantify uncertainty in the predicted scores, such as GPPL. GPPL also allows the inclusion of prior predictions as a prior mean, which helped both active learning and Bayesian optimisation in our experiments by providing information to identify the best candidates.

In future we intend to investigate techniques for integrating multiple sets of prior predictions into the model so that a selection of pre-trained models or heuristics can be used. This will also include pre-trained models that provide confidence estimates, which may help to kick-start Bayesian optimisation. We also plan to test our Bayesian optimisation approach on other NLP tasks that may suit interactive ranking, such as machine translation or natural language generation.

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.

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.

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.

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.

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.

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.

Armen Der Kiureghian and Ove Ditlevsen. 2009. Aleatory or epistemic? does it matter? *Structural Safety*, 31(2):105–112.

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.

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.

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.

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

Satish Reddy Indurthi, Seunghak Yu, Seohyun Back, and Heriberto Cuayahuitl. 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. Association for Comuptational Linguistics.

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

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.

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.

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.

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 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.

R. Duncan Luce. 1959. On the possible psychophysical laws. *Psychological Review*, 66(2):81–95.

Jonas Močkus. 1975. On bayesian methods for seeking the extremum. In *Optimization Techniques IFIP Technical Conference*, pages 400–404. Springer.

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.

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.

R. L. Plackett. 1975. The analysis of permutations. *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, 24(2):193–202.

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.

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.

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.

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. 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.

Burr Settles. 2012. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114.

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.

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

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*.

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.

Louis L. Thurstone. 1927. A law of comparative judgment. *Psychological Review*, 34(4):273–286.

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.

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.

Changchang Yin, Buyue Qian, Shilei Cao, Xiaoyu Li, Jishang Wei, Qinghua Zheng, and Ian Davidson. 2017. Deep similarity-based batch mode active learning with exploration-exploitation. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 575–584. IEEE.