Resubmission of TACL #1304
Finding Convincing Arguments using Scalable Bayesian Preference Learning.

January 12, 2018

Contents

1 Author(s) cover letter responding to the original reviews

Starts on next page.

Dear editors, reviewers,

Thank you for your insightful feedback for our submission number 1304. The reviews were very helpful in improving the quality of our paper. We have addressed the changes as follows:

- The SVM and BiLSTM methods were run with the combined feature set, 'ling+Glove', to enable a more direct comparison between methods. This required extending the BiLSTM model to accept document-level features. The new results support our proposed method as well as our claim that the combined feature set is effective for predicting convincingness.
- We analysed the scalability of our proposed method, with results for several tests shown in the section "Experiment 2: Scalability". This shows the effect of varying a key SVI hyperparameter, 'M', as well as the effect of data set size and number of features on runtimes. We compare the runtime scalability of our proposed inference method for GPPL (SVI) against a less scalable inference method as well as BiLSTM and SVM.
- We added a more detailed explanation of the GPPL model, as well as how SVI can be applied to this model to enable scalable inference. We have not included all of the equations for the complete SVI method as this would require too much space, but have aimed to give a complete and intuitive explanation of the method.
- We have made further minor changes to address as many of the reviewers' comments as possible, including clarifying some of our claims, modifying wording and typos.

2 Revised submission

Starts on next page.

Finding Convincing Arguments using Scalable Bayesian Preference Learning

First Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Second Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Abstract

We introduce a scalable Bayesian preference learning method for identifying convincing arguments in the absence of gold-standard ratings or rankings. In contrast to previous work, we avoid the need for separate methods to perform quality control on training data, predict rankings and perform pairwise classification. Bayesian approaches are an effective solution when faced with sparse or noisy training data, but have not previously been used to identify convincing arguments. One issue is scalability, which we address by developing a stochastic variational inference method for Gaussian process (GP) preference learning. We show how our method can be applied to predict argument convincingness from crowdsourced data, outperforming the previous state-of-the-art, particularly when trained with small amounts of unreliable data. We demonstrate how the Bayesian approach enables more effective active learning, thereby reducing the amount of data required to identify convincing arguments for new users and domains. While word embeddings are principally used with neural networks, our results show that word embeddings in combination with linguistic features also benefit GPs when predicting argument convincingness.

1 Introduction

Arguments are intended to persuade the audience of a particular point of view and are an important way for humans to reason about controversial topics (Mercier and Sperber, 2011). The amount of argumentative text on any chosen topic can, however, overwhelm a reader. Consider the scale of historical text archives and the debates on social media platforms with millions of users. Automated methods could enable readers to overcome this challenge by identifying high-quality, persuasive arguments from different sides of a debate.

Topic: "William Farquhar ought to be honoured as the rightful founder of Singapore".
Stance: "No, it is Raffles!"
Argument 1: HE HAS A BOSS(RAFFLES) HE HAS TO FOLLOW HIM AND NOT GO ABOUT DOING ANYTHING ELSE...
Argument 2: Raffles conceived a town plan to remodel Singapore into a modern city. The plan consisted of separate areas for different...
Crowdsourced labels: $\{2 \succ 1, 1 \succ 2, 2 \succ 1\}$

Figure 1: Example argument pair from an online debate.

Theoretical approaches for assessing argument quality have proved difficult to apply to everyday arguments (Boudry et al., 2015). However, empirical approaches using machine learning have recently shown success in identifying convincing arguments in online discussions (Habernal and Gurevych, 2016). Typically, machine learning approaches require examples of arguments paired with judgements of their convincingness. However, consider the arguments in Figure 1: how does one assign a numerical convincingness score to each argument? If the audience considers each argument independently, it is difficult to ensure that the scores for all arguments remain consistent with their view of convincingness.

An alternative way to judge arguments is to compare them against one another. In the case of arguments 1 and 2 in Figure 1, we may judge that argument 1 is less convincing due to its writing style, whereas argument 2 presents evidence in the form of historical events. Pairwise comparisons such as this are known to place less cognitive burden on human annotators than choosing a numerical rating and allow fine-grained sorting of items that is not possible with categorical labels (Kendall, 1948; Kingsley, 2006). Relative judgements also avoid the problem that different annotators may have biases toward high, low or middling ratings, making their scores hard to compare.

In practice, we face a data acquisition bottleneck when encountering new domains or audiences. For example, machine learning methods such as neural networks typically require datasets with many thousands of hand-labelled examples to perform well (Srivastava et al., 2014; Collobert et al., 2011). One solution is to employ multiple non-specialist annotators at low cost (*crowdsourcing*), but this requires quality control techniques to account for errors. Another source of data are the actions of users of a software application, which can be interpreted as pairwise judgements (Joachims, 2002). For example, when a user clicks on an argument in a list it can be interpreted as a preference for the selected argument over more highly-ranked arguments. However, the resulting pairwise labels are an extremely noisy indication of preference.

In this paper, we develop a Bayesian approach to learn from noisy pairwise preferences based on Gaussian process preference learning (GPPL) (Chu and Ghahramani, 2005). We model argument convincingness as a function of textual features, including word embeddings, and develop an inference method for GPPL that scales to realistic dataset sizes using a stochastic variational inference (SVI) (Hoffman et al., 2013). Using datasets provided by Habernal and Gurevych (2016), we show that our method outperforms the previous state-of-the-art for ranking arguments by convincingness and identifying the most convincing argument in a pair. Further experiments show that our Bayesian approach is particularly advantageous with small, noisy datasets, and in an active learning set-up. We make all of our experimental software and data publicly available at `http://github.com/******/******`.

The rest of the paper is structured as follows. Section 2 reviews related work on argumentation, then Section 3 motivates the use of Bayesian methods by discussing their successful applications in NLP. In Section 4, we review preference learning methods and then Section 5 describes our scalable Gaussian process-based approach. Section 6 presents our evaluation, comparing our method to the state-of-the art and testing with noisy data and active learning. Finally, we present conclusions and future work.

2 Identifying Convincing Arguments

Lukin et al. (2017) demonstrated that an audience's personality and prior stance affect an argument's persuasiveness, but they were unable to predict belief change to a high degree of accuracy. Related work has shown how persuasiveness is also affected by the sequence of arguments in a discussion (Tan et al., 2016; Rosenfeld and Kraus, 2016; Monteserin and Amandi, 2013), but this work focuses on predicting salience of an argument given the state of the debate, rather than the qualities of arguments.

Habernal and Gurevych (2016) established datasets containing crowdsourced pairwise judgements of convincingness for arguments taken from online discussions. Errors in the crowdsourced data were handled by determining gold labels using the MACE algorithm (Hovy et al., 2013). The gold labels were then used to train SVM and bi-directional long short-term memory (BiLSTM) classifiers to predict pairwise labels for new arguments. The gold labels were also used to construct a directed graph of convincingness, which was input to PageRank to produce scores for each argument. These scores were then used to train SVM and BiLSTM regression models. A drawback of such pipeline approaches is that they are prone to error propagation (Chen and Ng, 2016), and consensus algorithms such as MACE require multiple crowdsourced labels for each argument pair, which increases annotation costs.

3 Bayesian Methods for NLP

When faced with a lack of reliable annotated data, Bayesian approaches have a number of advantages. Bayesian inference provides a mathematical frame-

work for combining multiple observations with prior information. Given a model, M, and observed data, D, we apply Bayes' rule to obtain a posterior distribution over M, which can be used to make predictions about unseen data or latent variables:

$$P(M|D) = \frac{P(D|M)P(M)}{P(D)}, \quad (1)$$

where $P(D|M)$ is the likelihood of the data given M, and $P(M)$ is the model prior. If the dataset is large, the likelihood will dominate the posterior, but when D is small, the posterior will remain closer to the prior. Rather than learning a posterior distribution over M, neural network training typically selects model parameters that maximise the likelihood, so they are prone to overfitting with small datasets, which can reduce performance (Xiong et al., 2011).

Bayesian methods can be trained using unsupervised or semi-supervised learning to take advantage of structure in unlabelled data when labelled data is in short supply. Popular examples in NLP are Latent Dirichlet Allocation (LDA) (Blei et al., 2003), which is used for topic modelling, and its extension, the hierarchical Dirichlet process (HDP) (Teh et al., 2005), which learns the number of topics rather than requiring it to be fixed a priori. Semi-supervised Bayesian learning has also been used to achieve state-of-the-art results for semantic role labelling (Titov and Klementiev, 2012).

We can combine independent pieces of weak evidence using Bayesian methods through the likelihood. For instance, a Bayesian network can be used to infer attack relations between arguments by combining votes for acceptable arguments from different people (Hiroyuki Kido, 2017). Graphical models can also combine crowdsourced annotations to train a sentiment classifier without a separate quality control step (Simpson et al., 2015; Felt et al., 2016).

Several successful Bayesian approaches in NLP make use of Gaussian processes (GPs), which are distributions over functions of input features. GPs are nonparametric, meaning they can model highly nonlinear functions by allowing function complexity to grow with the amount of data (Rasmussen and Williams, 2006). They account for model uncertainty when extrapolating from sparse training data and can be incorporated into larger graphical models. Example applications include analysing the relationship between a user's impact on Twitter and the text features of their tweets (Lampos et al., 2014), predicting the level of emotion in text (Beck et al., 2014), and estimating the quality of machine translations given source and translated texts (Cohn and Specia, 2013).

4 Preference Learning

Our aim is to develop a Bayesian method for identifying convincing arguments given their features, which can be trained on noisy pairwise labels. Each label, $i \succ j$, states that an argument, i, is more convincing than another argument, j. This learning task is a form of *preference learning*, which can be addressed in several ways. A simple approach is to use a generic classifier by obtaining a single feature vector for each pair in the training and test datasets, either by concatenating the feature vectors of the items in the pair or by computing the difference of the two feature vectors, as in SVM-Rank (Joachims, 2002). However, this approach does not produce ranked lists of convincing arguments without predicting a large number of pairwise labels, nor give scores of convincingness.

Alternatively, we can learn an ordering over arguments directly using Mallows models (Mallows, 1957), which define distributions over list permutations. Mallows models can be trained from pairwise preferences (Lu and Boutilier, 2011), but inference is typically costly since the number of possible permutations is $\mathcal{O}(N^2)$, where N is the number of arguments. Modelling only the ordering does not allow us to quantify the difference between arguments at similar ranks.

To avoid the problems of classifier-based and permutation-based methods, we propose to learn a real-valued convincingness function, f, that takes argument features as input and can be used to predict rankings, pairwise labels, or ratings for individual arguments. There are two well established approaches for mapping pairwise labels to real-valued scores: the Bradley-Terry-Plackett-Luce model (Bradley and Terry, 1952; Luce, 1959; Plackett, 1975) and the Thurstone-Mosteller model (Thurstone, 1927; Mosteller, 2006). Based on the latter approach, Chu and Ghahramani (2005) introduced Gaussian process preference learning (GPPL), a

Bayesian model that can tolerate errors in pairwise training labels and gains the advantages of a GP for learning nonlinear functions from sparse datasets. However, the inference method proposed by Chu and Ghahramani (2005) has memory and computational costs that scale with $\mathcal{O}(N^3)$, making it unsuitable for real-world text datasets. The next section explains how we use recent developments in inference methods to develop scalable Bayesian preference learning for argument convincingness.

5 Scalable Bayesian Preference Learning

First, we define a probabilistic model for preference learning (Chu and Ghahramani, 2005). We observe preference pairs, each consisting of a pair of feature vectors $\mathbf{x}_i$ and $\mathbf{x}_j$, for arguments i and j, and a label $y \in \{i \succ j, j \succ i\}$, where $i \succ j$ indicates that i is more convincing than j, and $j \succ i$ means the opposite. We assume that the likelihood of y depends on the latent convincingness, $f(\mathbf{x_i})$ and $f(\mathbf{x}_j)$, of the arguments in the pair. Our goals is to predict y for pairs that have not been observed, and $f(\mathbf{x}_i)$, which may be used to rank arguments.

The relationship between convincingness and pairwise labels is described by the following:

$$p(x_i \succ x_j | f(x_i), f(x_j), \delta_i, \delta_j) = \begin{cases} 1 & \text{if } f(x_i) + \delta_i \geq f(j) + \delta_j \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

where $\delta \sim \mathcal{N}(0, 1)$ is Gaussian-distributed noise. If the convincingness $f(x_i)$ is higher than the convincingness $f(x_j)$, the preference label $i \succ j$ is more likely to be true. However, the label also depends on the noise terms, δ_i and δ_j, to allow for errors caused by, for example, disagreement between human annotators. We simplify Equation 2 by integrating out δ_i and δ_j to obtain the *preference likelihood*:

$$\begin{aligned} &p(x_i \succ x_j | f(x_i), f(x_j)) \\ &= \int\int p(x_i \succ x_j | f(x_i), f(x_j), \delta_i, \delta_j) \\ &\qquad \mathcal{N}(\delta_i; 0, 1)\mathcal{N}(\delta_j; 0, 1) d\delta_i d\delta_j \\ &= \Phi(z), \end{aligned} \quad (3)$$

where $z = (f(x_i) - f(x_j))/\sqrt{2}$ and Φ is the cumulative distribution function of the standard normal distribution.

We assume that convincingness is a function, f, of argument features, drawn from a Gaussian process prior: $f \sim \mathcal{GP}(0, k_\theta s)$, where k_θ is a kernel function with hyper-parameters θ, and s is a scale parameter. The kernel function controls the smoothness of f over the feature space, while s controls the variance of f. Increasing s means that, on average, the magnitude of $f(x_i) - f(x_j)$ increases so that $\Phi(z)$ is closer to 0 or 1, and erroneous pairwise labels are less likely. Therefore, larger values of s correspond to lower noise levels. We place a Gamma distribution over $1/s$ with shape a_0 and scale b_0, as this is a conjugate prior: $1/s \sim \mathcal{G}(a_0, b_0)$.

Given a set of N arguments and P labelled preference pairs, $\mathbf{y} = \{y_1, ..., y_P\}$, we can make predictions by finding the posterior distribution over the argument convincingness values, $\mathbf{f} = \{f(\mathbf{x}_1), ..., f(\mathbf{x}_N))$, given by:

$$p(\mathbf{f}|\mathbf{y}, k_\theta, a_0, b_0) = \frac{1}{Z} \int \prod_{k=1}^{P} \Phi(z_k) \mathcal{N}(\boldsymbol{f}; \mathbf{0}, \boldsymbol{K}_\theta s) \mathcal{G}(s; a_0, b_0) ds, \quad (4)$$

where $Z = p(\mathbf{y}|k_\theta, a_0, b_0)$ is the normalization constant. Unfortunately, neither Z nor the integral over s can be computed analytically, so we must turn to approximations.

Chu and Ghahramani (2005) used a Laplace approximation for GPPL, which finds a maximum a-posteriori (MAP) solution that has been shown to perform poorly in many cases (Nickisch and Rasmussen, 2008). More accurate estimates of the posterior could be obtained using Markov chain Monte Carlo sampling (MCMC), but this is very computationally expensive (Nickisch and Rasmussen, 2008). Instead, we use a faster *variational* method that maintains the benefits of the Bayesian approach (Reece et al., 2011; Steinberg and Bonilla, 2014) and adapt this method to the preference likelihood given by Equation 3.

To apply the variational approach, we define an approximation $q(\mathbf{f})$ to Equation 4. First, we approximate the preference likelihood with a Gaussian, $\prod_{k=1}^{P} \Phi(z_k) \approx \mathcal{N}(\mathbf{y}; \mathbf{G}\hat{\mathbf{f}}, \mathbf{Q})$. This allows us to avoid the intractable integral in Z and obtain another Gaussian, $q(\boldsymbol{f}) = \mathcal{N}(\mathbf{f}; \hat{\mathbf{f}}, \mathbf{C})$. The parameters $\hat{\mathbf{f}}$ and $\mathbf{C}$ depend on the approximate preference likelihood and an approximate distribution over s: $q(s) =$

$\mathcal{G}(s; a, b)$. The variational inference algorithm begins by initialising the parameters $\mathbf{G}$, $\hat{\mathbf{f}}$, $\mathbf{C}$, a and b at random. Then, the algorithm proceeds iteratively updating each parameter in turn, given the current values for the other parameters. This optimisation procedure minimises the Kullback-Leibler (KL) divergence of $p(\mathbf{f}|\mathbf{y}, k_\theta, a_0, b_0)$ from $q(f)$, causing $q(f)$ to converge to an approximate posterior.

The update equations for the mean $\hat{\mathbf{f}}$ and covariance $\mathbf{C}$ require inverting the covariance matrix, K_θ, at a computational cost of $\mathcal{O}(N^3)$, which is impractical with more than a few hundred data points. Furthermore, the updates also require $\mathcal{O}(NP)$ computations and have $\mathcal{O}(N^2 + NP + P^2)$ memory complexity. To resolve this, we apply a recently introduced technique, stochastic variational inference (SVI) (Hoffman et al., 2013; Hensman et al., 2015), to scale to datasets containing at least tens of thousands of arguments and pairwise labels.

SVI makes two approximations: it assumes M *inducing points*, which act as a substitute for the observed arguments; it uses only a random subset of the data containing P_n pairs at each iteration. At each iteration, t, rather than updates to $\hat{\mathbf{f}}$ and $\boldsymbol{C}$ directly, we update the mean $\hat{\mathbf{f}_m}$ and covariance $\boldsymbol{C}_m$ for the inducing points. The updates for each parameter $\lambda \in \{\hat{\mathbf{f}_m}, \boldsymbol{C}_m\}$ takes the form of a weighted average between the previous estimate and a new estimate computed from only a subset of observations:

$$\lambda^{(t)} = (1 - \rho_t)\lambda^{(t-1)} + \rho_t \hat{\lambda}_t P/P_n, \tag{5}$$

where $\rho = t^{-u}$ is the step size, u is a forgetting rate,and $\hat{\lambda}_t$ is the new estimate computed from P_n out of P observations. The values of $\hat{\mathbf{f}}$ and $\boldsymbol{C}$ can be estimated from the inducing point distribution. By choosing $M << N$ and $P_n << P$, we limit the computational complexity of each SVI iteration to $\mathcal{O}(M^3 + MP_n)$ and the memory complexity $\mathcal{O}(M^2 + MP_n + P_n^2)$. To choose representative inducing points, we use K-means with $K = M$ to rapidly cluster the feature vectors, then take the cluster centres as inducing points.

A further benefit of GPs is that they enable automatic relevance determination (ARD) to identify informative features, which works as follows. The prior covariance of f is defined by a kernel function of the form $k_\theta(\mathbf{x}, \mathbf{x}') = \prod_{d=1}^{D} k_d(|x_d - x'_d|/l_d)$, where k_d is a function of the distance between the values of feature d for items x and x', and a length-scale hyper-parameter, l_d. The length-scale controls the smoothness of the function across the feature space, and can be optimised by choosing the value of l_d that maximises the approximate log marginal likelihood, $\mathcal{L} \approx \log p(\boldsymbol{y})$. This process is known as *maximum likelihood II* (Rasmussen and Williams, 2006). Features with larger length-scales after optimisation are less relevant because their values have less effect on $k_\theta(\mathbf{x}, \mathbf{x}')$. To cut out the cost of optimising the length-scales, we can alternatively set them using a median heuristic, which has been shown to perform well in practice (Gretton et al., 2012): $l_{d,MH} = \frac{1}{D}\text{median}(\{|x_{i,d} - x_{j,d}|\forall i = 1, .., N, \forall j = 1, ..., N\})$.

6 Experiments

6.1 Datasets

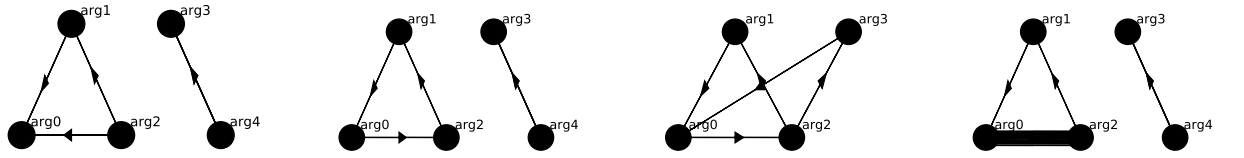
We first use toy datasets to illustrate the behaviour of several different methods (described below). Then, we analyse the scalability and performance our approach on datasets provided by Habernal and Gurevych (2016), which contain pairwise labels for arguments taken from online discussion forums. Labels can have a value of 0, meaning the annotator found the second argument in the pair more convincing, 1 if the annotator was undecided, or 2 if the first argument was more convincing. To test different scenarios, different pre-processing steps were used to produce the three *UKPConvArg** datasets shown in Table 1. For these datasets we perform 32-fold cross validation, using 31 folds for training and one for testing. Each fold corresponds to one of 16 controversial topics, and one of two stances for that topic. *UKPConvArgStrict* and *UKPConvArgRank* test performance with noise-free labelled data, while *UKPConvArgCrowdSample* is used to evaluate performance with noisy crowdsourced data including conflicts and undecided labels, and to test the suitability of our method for active learning to address the cold-start problem in new domains with no labelled data.

6.2 Method Comparison

Our two basic tasks are *ranking* arguments by convincingness and *classification* of pairwise labels for

Dataset	Pairs	Arguments	Undecided	Dataset properties
Toy Datasets	4-13	4-5	0-9	Synthetic pairwise labels Arguments sampled at random from UKPConvArgStrict
UKPConvArg-Strict	11642	1052	0	Combine crowdsourced pairwise labels with MACE Gold labels are $\geq 95\%$ most confident MACE labels Discard arguments marked as equally convincing Discard conflicting preferences
UKPConvArg-Rank	16081	1052	3289	Combine crowdsourced pairwise labels with MACE Gold labels are $\geq 95\%$ most confident MACE labels PageRank run on each topic to produce gold rankings
UKPConvArg-CrowdSample	16927	1052	3698	One original crowdsourced label per pair PageRank run on each topic to produce gold rankings Labels for evaluation from UKPConvArgStrict/UKPConvArgRank

Table 1: Summary of datasets, showing the different steps used to produce each Internet argument dataset.



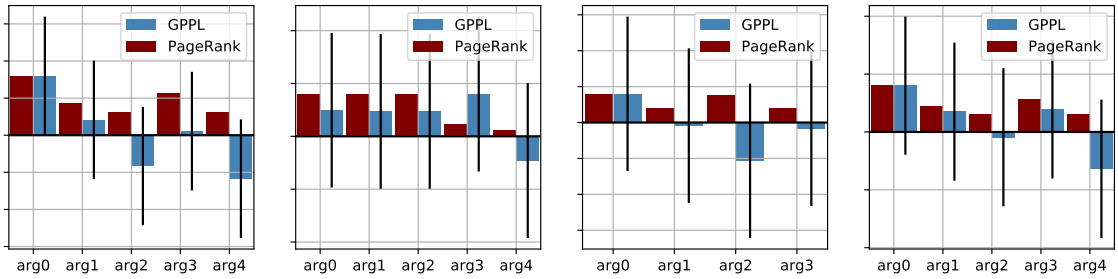
(a) no cycle (b) single cycle (c) double cycle (d) no cycle + 9 undecided prefs.

Figure 2: Argument preference graphs for each scenario. Arrows point to the preferred argument.

pairs of arguments, i.e. predicting which argument is more convincing. We compare our scalable Gaussian process preference learning method (*GPPL*) against an SVM approach with radial basis function kernels, and a bi-directional long short-term memory network (BiLSTM), with 64 output nodes in the core LSTM layer. Both methods were tested by Habernal and Gurevych (2016) and are available in our software repository. For both the classification and ranking tasks, GPPL is trained using the pairwise labels for the training folds. We rank arguments by their expected convincingness, $\mathbb{E}[f(\mathbf{x}_n)] \approx \hat{f}_n]$ for each argument i with feature vector $\mathbf{x}_n$, under the approximate posterior $q(\boldsymbol{f})$ output by our SVI algorithm. Classification probabilities are obtained by substituting values of $\hat{f}_n$ for corresponding $f(i)$ and $f(j)$ terms in Equation 3. To apply SVM and BiLSTM to the classification task, we concatenate the feature vectors of each pair of arguments in the training and test sets, and train on the pairwise labels. For ranking, PageRank is first applied to arguments in the training folds to obtain gold-standard scores from the pairwise labels. SVM and BiLSTM regression models are then trained using the PageRank scores.

As a Bayesian alternative to GPPL, we test a Gaussian process classifier (*GPC*) for the classification task by concatenating the feature vectors of arguments in the same way as the SVM classifier. We also evaluate a non-Bayesian approach that uses the same pairwise likelihood as GPPL (Equation 3) to infer function values, but uses them to train an SVM regression model instead of a GP (*PL+SVR*).

We use two sets of input features. The *ling* feature set contains 32010 linguistic features, including unigrams, bigrams, ratios and counts of different parts-of-speech and verb forms, dependency tree depth, ratio of exclamation or quotation marks, counts of several named entity types, POS n-grams, presence of dependency tree production rules, readability measures, sentiment scores, spell-checking, and word counts. The *Glove* features are word embeddings with 300 dimensions. Both of these feature sets were developed by Habernal and Gurevych (2016). We also evaluate a combination of both feature sets, *ling + Glove*. To create



(a) no cycle (b) single cycle (c) double cycle (d) cycle with 9 undecided

Figure 3: Mean scores over 25 repeats. Bars for GPPL show standard deviation of convincingness function posterior.

a single embedding vector per argument as input for GPPL, we take the mean of individual word embeddings for tokens in the argument. We also tested skip-thoughts (Kiros et al., 2015) and Siamese-CBOW (Kenter et al., 2016) with GPPL on UKPConvArgStrict and UKPConvArgRank, both with MLII optimisation and the median heuristic, both alone and combined with *ling*. However, we found that mean Glove embeddings produced substantially better performance in all tests. When running BiLSTM with *ling* features, we include an additional dense layer with 64 nodes.

We set the GPPL hyper-parameters $a_0 = 2$ and $b_0 = 200$ by comparing training set performance on UKPConvArgStrict and UKPConvArgRank against $a_0 = 2$, $b_0 = 20000$ and $a_0 = 2$, $b_0 = 2$. The chosen prior is very weakly informative, favouring a moderate level of noise in the pairwise labels. For the kernel function, k_d, we tested only the Matérn $\frac{3}{2}$ function due to its effectiveness across a wide range of tasks (Rasmussen and Williams, 2006). To set length-scales, l_d, we compare the median heuristic (labelled "medi.") with MLII optimisation using the L-BFGS optimisation algorithm ("opt."). Experiment 2 shows how the number of inducing points, M, can be set to trade off speed and accuracy. Following those results, we set $M = 500$ for Experiments 3, 4 and 5 and $M = N$ for the small toy dataset in Experiment 1.

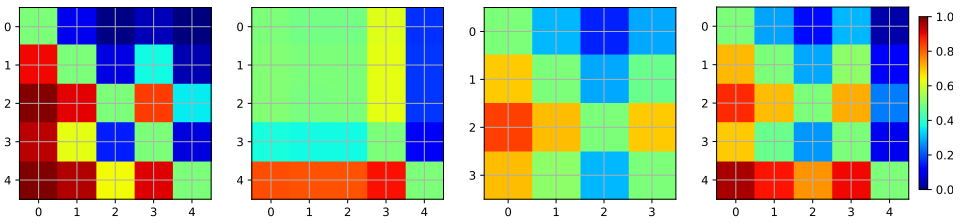
6.3 Experiment 1: Toy Data

We use synthetic data to illustrate the different behaviour of GPPL, SVM for pairwise classification, and PageRank for scoring arguments. We simulate four scenarios, each of which contains arguments labelled *arg0* to *arg4*. In each scenario, we generate a set of pairwise preference labels according to the convincingness graphs shown in Figure 2. Each scenario is repeated 25 times: in each repeat, we select arguments at random from one fold of UKPConvArgStrict then associate the mean Glove embeddings for these arguments with the labels arg0 to arg4. We train GPPL, PageRank and the SVM classifier on the preference pairs shown in each graph and make predictions for arguments arg0 to arg4.

In the "no cycle" scenario, arg0 is preferred to both arg1 and arg2, which is reflected in the PageRank and GPPL scores in Figure 3. However, arg3 and arg4 are not connected to the rest of the graph and receive different scores with PageRank and GPPL. Figure 4 shows how GPPL provides probabilistic classifications that are less confident for pairs that were not yet observed, e.g. arg2 $\succ$ arg4. This contrasts with Figure 5 which shows discrete classifications produced by SVM.

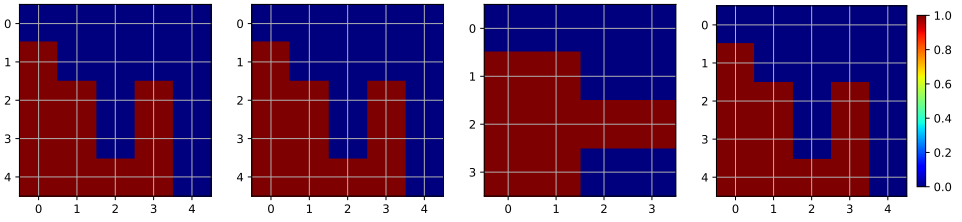
The "single cycle" scenario shows how each method handles a cycle in the preference graph. Both PageRank and GPPL produce equal values for the arguments in the cycle (arg0, arg1, arg2). PageRank assigns lower scores to both arg3 and arg4 than the arguments in the cycle, while GPPL more intuitively gives a higher score to arg3, which was preferred to arg4. SVM predicts that arg0 and arg1 are preferred over arg3, although arg0 and arg1 are in a cycle so there is no reason to prefer them. GPPL, in contrast, weakly predicts that arg3 is preferred.

In the "double cycle" scenario, PageRank and GPPL produce very different results. Here, the argument graph shows two paths from arg2 to arg0 via arg1 or arg3, and one conflicting preference arg2



(a) no cycle (b) single cycle (c) double cycle (d) cycle with 9 undecided

Figure 4: Mean GPPL predictions over 25 repeats. Probability that the argument on the horizontal axis is preferred to the argument on the vertical axis.



(a) no cycle (b) single cycle (c) double cycle (d) cycle with 9 undecided

Figure 5: Mean SVM predictions over 25 repeats. Probability that the argument on the horizontal axis is preferred to the argument on the vertical axis.

$\succ$ arg. GPPL scores the arguments as if the single conflicting preference, arg2 $\succ$ arg0, is less important than the two parallel paths from arg2 to arg. In contrast, PageRank gives high scores to both arg0 and arg2. The classifications by GPPL and SVM are similar, but GPPL produces more uncertain predictions than in the first scenario due to the conflict.

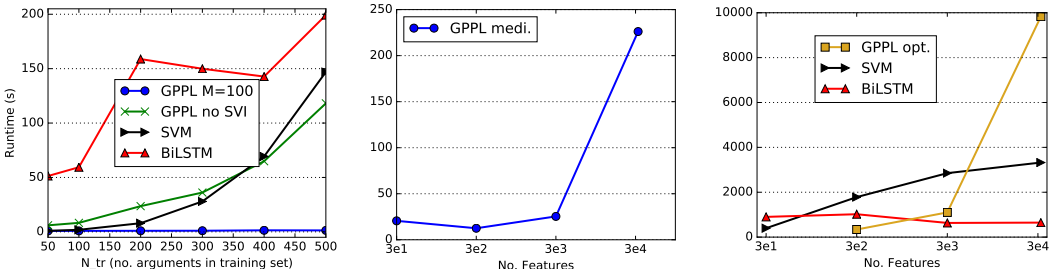
Finally, Figure 3d shows the addition of nine undecided labels to the "no cycle" scenario, indicated by undirected edges in Figure 2, to simulate a case where multiple annotators labelled the pair. This does not affect the PageRank scores, but reduces the difference in GPPL scores between arg0 and the other arguments, since GPPL gives the edge from arg2 to arg0 less weight due to the undecided labels. This is reflected in the GPPL classifications, which are less confident than in the "no cycle" scenario. The SVM cannot be trained using uncertain labels and therefore does not adapt to the undecided labels.

In conclusion, GPPL appears to resolve conflicts in the preference graphs in a more intuitive manner than PageRank, which was designed for ranking web pages by importance rather than preference. In contrast to SVM, GPPL is able to account for undecided labels to soften the latent convincingness function.

6.4 Experiment 2: Scalability

We analyse empirically the scalability of the proposed SVI method for GPPL using the UKPConvArgStrict dataset. Figure 7 shows the effect of varying the number of inducing points, M, on the overall runtime and accuracy of the method. The accuracy increases quickly with M, and flattens out, suggesting there is little benefit to increasing M further on this dataset. The runtimes increase with M, and are much longer with 32310 features than with 300 features. The difference is due to the cost of computing the kernel, which is linear in M, With only 300 features, Figure 7a appears polynomial, reflecting the $\mathcal{O}(M^3)$ term in the inference procedure.

We tested GPPL with both the SVI algorithm, with $M = 100$ and $P_n = 200$, and variational inference without inducing points or stochastic updates (labelled "no SVI") with different sizes of training dataset subsampled from UKPConvArgStrict. The results are shown in Figure 6a. For GPPL with SVI, the runtime increases very little with dataset size, while the runtime with "no SVI" increases polynomially with training set size (both N and P). At $N = 100$, the number of inducing points is $M = N$ but the SVI algorithm is still faster due

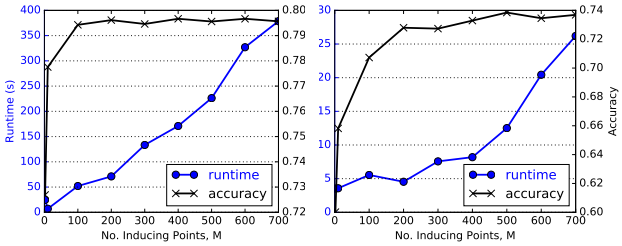


(a) Varying no. arguments in training set, Glove features

(b) Varying no. ling+Glove features, GPPL, medi., M=500

(c) Varying no. ling+Glove features, long-running methods

Figure 6: Runtimes for training+prediction on UKPConvArgStrict with different subsamples of data. Means over 32 runs. Note logarithmic x-axis for (b) and (c).



(a) 33210 ling+Glove features

(b) 300 Glove features

Figure 7: Effect of varying M on accuracy and runtime (training+prediction) of GPPL for UKPConvArgStrict. Means over 32 runs.

to the stochastic updates with $P_n = 200$ rather than $P > 200$ pairs.

Figure 6b shows the effect of the number of features, D, on runtimes. Runtimes for GPPL increase by a large amount with $D = 32310$, because the SVI method computes the kernel matrix, $K_m m$, with computational complexity $\mathcal{O}(D)$. While D is small, other costs dominate. We show runtimes using the MLII optimisation procedure with GPPL in Figure 6c. The procedure was run with a maximum of 25 iterations and did not terminate in fewer than 25 in any of the test runs. This creates a similar pattern to Figure 6b (approximately multiples of 50). Owing to the long computation times required, we were unable to test the optimisation procedure to convergence.

We include runtimes for SVM and BiLSTM in Figures 6a and 6c to show their runtime patterns, but note that the runtimes reflect differences in implementations and system hardware. Both SVM and GPPL were run on an Intel i7 quad-core desktop. For SVM we used LibSVM version 3.2, which could be sped up if probability estimates were not required. BiLSTM was run with Theano 0.7[1] on an Nvidia Tesla P100 GPU. We can see in Figure 6c that the runtime for BiLSTM does not appear to increase due to the number of features, while that of SVM increases sharply with 32310 features. In Figure 6a, we observe the SVM runtimes increase polynomially with training set size.

6.5 Experiment 3: UKPConvArgStrict and UKPConvArgRank

We compare classification performance on UKPConvArgStrict and ranking performance on UKPConvArgRank. Both datasets were cleaned to remove disagreements between annotators, as stated in Table 1, hence can be considered to be *noise-free*. The results in Table 2 show that when using *ling* features, GPPL produces similar accuracy and improves the area under the ROC curve (AUC) by .02 and cross entropy error (CEE) by .01. AUC quantifies how well the predicted probabilities separate the classes, while CEE quantifies the usefulness of the probabilities output by each method. Much larger improvements can be seen in the ranking metrics. When GPPL is run with *Glove*, it performs worse than BiLSTM for classification but improves the ranking metrics. Using a combination of features improves all methods, suggesting that embeddings and linguistic features contain complementary

[1] http://deeplearning.net/software/theano/

	SVM		BiLSTM		GPPL medi.			GPPL opt.	GPC	PL+ SVR
	ling	ling +Glove	Glove	ling +Glove	ling	Glove	ling +Glove			
UKPConvArgStrict (pairwise classification)										
Acc.:	.78	.79	.76	.77	.78	.71	.79	.80	**.81**	.78
AUC:	.83	.86	.84	.86	.85	.77	.87	.87	**.89**	.85
CEE:	.52	.47	.64	.57	.51	1.12	.47	.51	**.43**	.51
UKPConvArgRank (ranking)										
Pearson's r:	.36	.37	.32	.36	.38	.33	**.45**	.44	-	.39
Spearman's ρ:	.47	.48	.37	.43	.62	.44	.65	**.67**	-	.63
Kendall's τ:	.34	.34	.27	.31	.47	.31	.49	**.50**	-	.47

Table 2: Performance comparison on UKPConvArgStrict and UKPConvArgRank datasets.

information.

Optimising the length-scale using MLII improves classification accuracy by 1% over the median heuristic, and significantly improves accuracy ($p = .043$ using two-tailed Wilcoxon signed-rank test) and AUC ($p = .013$) over the previous state-of-the-art, SVM with linguistic features. The differences in all of the ranking metrics between GPPL opt. and the next-best methd, SVM with *ling + Glove*, are statistically significant, with $p = .029$ for Pearson's r and $p <<< .01$ for both Spearman's ρ and Kendall's τ. However, the cost of these improvements is that each fold required around 2 hours to compute instead of approximately 10 minutes on the same machine (an Intel i7 quad-core desktop) using the median heuristic.

GPC produces the best results on the classification task (with $p < 0.01$ for all metrics compared to all other methods), indicating the benefits of a Bayesian approach over SVM and BiLSTM. However, unlike GPPL, GPC cannot be used to rank the arguments. The results also show that PL+SVR does not reach the same performance as GPPL, suggesting that GPPL may benefit from the Bayesian integration of a GP with the preference likelihood.

6.6 Experiment 4: Conflicting and Noisy Data

We use UKPConvArgCrowdSample to introduce noisy data and conflicting pairwise labels, to both the classification and regression tasks, to test the hypothesis that GPPL would best handle unreliable crowdsourced data. The evaluation uses the labels from UKPConvArgStrict and UKPConvArgRank for items in the test set. The results in Table 3 show that all methods perform worse compared to Experiment 3 due to the presence of errors in the pairwise labels in UKPConvArgCrowdSample. Here, GPPL produces the best classification accuracy and cross-entropy error (significant with $p <<< .01$ compared to all other methods except accuracy compared to GP+SVR, for which $p = .045$), while GPC has the highest AUC. Compared to UKPConvArgStrict, the classification performance of GPC, SVM and BiLSTM have decreased more than that of GPPL. These methods lack a mechanism to resolve conflicts in the preference graph, unlike GPPL and PL+SVR, which handle conflicts through the preference likelihood. PL+SVR again performs worse than GPPL on classification metrics, although its ranking performance is comparable. For ranking, GPPL again outperforms SVM and BiLSTM in all metrics (significant with $p << .01$ in all cases except for SVM with Pearson's correlation).

	SVM	Bi-LSTM	GPPL medi.	PL+ SVR	GPC
Classification					
Acc	.70	.73	**.77**	.75	.73
AUC	.81	.81	.84	.82	**.86**
CEE	.58	.55	**.50**	.55	.53
Ranking					
Pears.	.32	.22	**.35**	.31	-
Spear.	.43	.30	.54	**.55**	-
Kend.	.31	.21	**.40**	**.40**	-

Table 3: Performance comparison on the UKPConvArgCrowdSample datasets containing conflicts and noise using ling+Glove features.

6.7 Experiment 5: Active Learning

In this experiment, we hypothesised that GPPL provides more meaningful confidence estimates than SVM or BiLSTM, which can be used to facilitate active learning in scenarios where labelled training data is expensive or initially unavailable. To test this hypothesis, we simulated an active learning scenario, in which an agent iteratively learns a model for each fold. Initially, $N_{inc} = 2$ pairs were chosen at random, then used to train the classifier. The agent then performs *uncertainty sampling* (Settles, 2010) to select the $N_{inc} = 2$ pairs with the least confident classifications. The labels for these pairs are then added to the training set and used to re-train the model. The process was repeated until 400 labels had been sampled.

The result is plotted in Figure 8, showing that GPPL reaches a mean accuracy of 70% with only 100 labels, while SVM and BiLSTM do not reach the same performance given 400 labels. After 100 labels, the performance of BiLSTM decreases. It has previously been shown (Cawley, 2011; Guyon et al., 2011; Settles, 2010) that uncertainty sampling can perform poorly in some cases, leading to periods where accuracy decreases as labels are received. Such a failure may be more likely when a model overfits to the current, small set of samples, causing it to mis-classify some data points with high confidence, meaning the mis-classified points will not be selected. The larger number of parameters in the BiLSTM means that it may be more prone to overfitting with small datasets than SVM or GPPL. The results suggest that GPPL may outperform the alternatives in cold-start scenarios where there are initially small amounts of labelled data, since the Bayesian approach reduces overfitting by accounting for parameter uncertainty.

6.8 Relevant Feature Determination

Finally, we show how the length-scales learned by optimising GPPL using MLII can be used to identify informative sets of features. A larger length-scale causes greater smoothing, implying that the feature is less relevant when predicting the convincingness function than a feature with a small length-scale. Figure 9 shows the distribution of normalised length-scales for *ling+Glove* after optimising on one

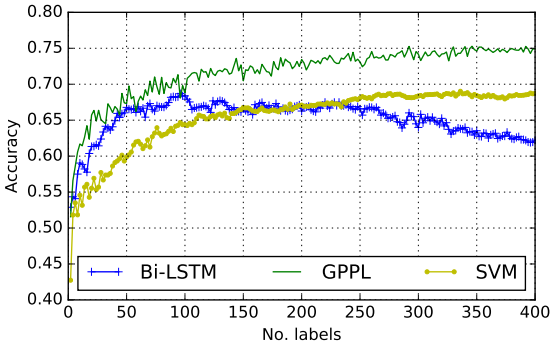


Figure 8: Active learning simulation showing mean accuracy of preference pair classifications over 32 runs.

fold of UKPConvArgStrict. Due to the computation time required, the optimisation algorithm was limited to 25 iterations, resulting in the large number of values close to 1, as features with larger gradients were optimised first. The length-scales for many dimensions of the mean word embeddings were increased, giving ratios close to 4 times the median heuristic, suggesting that these dimensions may be only very weakly informative in comparison to other dimensions of the embedding vectors. Table 4 shows the largest and smallest ratios for embeddings and linguistic features. The unigram "safety" has a very high length-scale, suggesting it is not informative and could be discarded. The length-scales learned using MLII could in future be used to discard irrelevant features automatically.

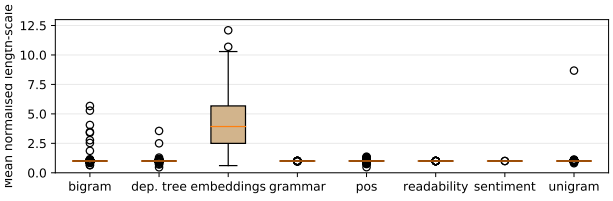


Figure 9: Distribution of length-scales for each type of feature after MLII optimisation on fold "should physical education be mandatory in schools – no". Values are ratios over the median heuristic value. Optimisation on this fold increased accuracy from 75% to 80%.

6.9 Error Analysis

We compared the errors when using GPPL opt. with mean Glove embeddings and with linguistic features. We manually inspected the twenty-five arguments most frequently mis-classified by GPPL *ling*

Feature	Ratio
ProductionRule-S->ADVP,NP,VP,.,	.466
Pos-ngram-PP-O-CARD	.477
Unigram-“safer”,	.640
Bigram-“?”-“look”	5.672
Unigram-“safest”	8.673
Unigram-“safety”	271.190
Embedding-dimension-19	.610
Embedding-dimension-241	12.093

Table 4: Ratios of optimised to median heuristic length-scales: largest and smallest ratios for linguistic features and word embeddings.

and correctly classified by GPPL *Glove*. We found that GPPL *ling* mistakenly marked several arguments as less convincing when they contained grammar and spelling errors but otherwise made a logical point. In contrast, arguments that did not strongly take a side and did not contain language errors were often marked mistakenly as more convincing.

We also examined the twenty-five arguments most frequently misclassified by GPPL *Glove* but not by GPPL *ling*. Of the arguments that GPPL *Glove* incorrectly marked as more convincing, 10 contained multiple exclamation marks and all-caps sentences. Other failures were very short arguments and underrating arguments containing the term ‘rape’. The analysis suggests that the different feature sets identify different aspects of convincingness.

To investigate the differences between our best approach, GPPL opt. *ling* + *Glove*, and the previous best performer, SVM, we manually examined forty randomly chosen false classifications, where one of either *ling* + *Glove* or SVM was correct and the other was incorrect. We found that both SVM and GPPL falsely classified arguments that were either very short or long and complex, suggesting deeper semantic or structural understanding of the argument may be required. However, SVM also made mistakes where the arguments contained few verbs.

We also compared the rankings produced by GPPL opt. (ling+Glove), and SVM on UKPConvArgRank by examining the 20 largest deviations from the gold standard rank for each method. Arguments underrated by SVM and not GPPL often contained exclamation marks or common spelling errors (likely due to unigram or bigram features). GPPL underrated short arguments with the ngrams “I think”, “why?”, and “don’t know”, which were used as part of a rhetorical question rather than to state that the author was uncertain or uninformed. These cases may not be distinguishable by a GP given only *ling* + *Glove* features.

An expected advantage of GPPL is that it provides more meaningful uncertainty estimates for tasks such as active learning. We examined whether erroneous classifications correspond to more uncertain predictions when using GPPL and SVM when both methods use the *ling* features. For UKPConvArgStrict, the mean Shannon entropy of the pairwise predictions from GPPL was .129 for correct predictions and 2.443 for errors, while for SVM, the mean Shannon entropy was .188 for correct predictions and 1.583 for incorrect. With both methods, more uncertain predictions correlate with more errors, but the more extreme values for GPPL suggest that its output probabilities more accurately reflect the probability of error than those given by the SVM.

7 Conclusions and Future Work

We presented a novel Bayesian approach to predicting argument convincingness from pairwise labels using Gaussian process preference learning (GPPL). Using recent advances in approximate inference, we developed a scalable algorithm for GPPL that is suitable for large NLP datasets. Our experiments demonstrated that our method significantly outperforms the state-of-the-art on a benchmark dataset for argument convincingness, particularly when noisy and conflicting pairwise labels are used in training. The active learning results show that GPPL is an effective model for cold-start situations and that the convincingness of Internet arguments can be predicted reasonably well given only a small number of samples. The results also showed that linguistic features and word embeddings provide complementary information, and that GPPL can be used to automatically identify relevant features.

Future work will evaluate our approach on other NLP tasks where reliable classifications may be difficult to obtain, such as learning to classifiy text from implicit user feedback (Joachims, 2002). We also plan to investigate whether the GP can be trained using absolute scores in combination with pairwise labels.

References

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 (EMNLP)*, pages 1798–1803. ACL.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent Dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Maarten Boudry, Fabio Paglieri, and Massimo Pigliucci. 2015. The fake, the flimsy, and the fallacious: demarcating arguments in real life. *Argumentation*, 29(4):431–456.

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.

Gavin C Cawley. 2011. Baseline methods for active learning. In *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*, pages 47–57.

Chen Chen and Vincent Ng. 2016. Joint inference over a lightly supervised information extraction pipeline: Towards event coreference resolution for resource-scarce languages. In *Thirtieth AAAI Conference on Artificial Intelligence*, pages 2913–2920.

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 *ACL (1)*, pages 32–42.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Paul Felt, Eric K Ringger, and Kevin D Seppi. 2016. Semantic annotation aggregation with conditional crowdsourcing models and word embeddings. In *COLING*, pages 1787–1796.

Arthur Gretton, Dino Sejdinovic, Heiko Strathmann, Sivaraman Balakrishnan, Massimiliano Pontil, Kenji Fukumizu, and Bharath K Sriperumbudur. 2012. Optimal kernel choice for large-scale two-sample tests. In *Advances in neural information processing systems*, pages 1205–1213.

Isabelle Guyon, Gavin Cawley, Gideon Dror, and Vincent Lemaire. 2011. Results of the active learning challenge. In *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*, pages 19–45.

Ivan Habernal and Iryna Gurevych. 2016. Which argument is more convincing? Analyzing and predicting convincingness of Web arguments using bidirectional LSTM. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1589–1599, Berlin, Germany. Association for Computational Linguistics.

James Hensman, Alexander G. de G. Matthews, and Zoubin Ghahramani. 2015. Scalable Variational Gaussian Process Classification. In *AISTATS*.

Keishi Okamoto Hiroyuki Kido. 2017. A Bayesian approach to argument-based reasoning for attack estimation. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 249–255.

Matthew D Hoffman, David M Blei, Chong Wang, and John William Paisley. 2013. Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard H Hovy. 2013. Learning whom to trust with MACE. In *Proceedings of NAACL-HLT 2013*, pages 1120–1130.

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.

Tom Kenter, Alexey Borisov, and Maarten de Rijke. 2016. Siamese CBOW: optimizing word embeddings for sentence representations. In *Proceedings of the The 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*.

David C Kingsley. 2006. Preference uncertainty, preference refinement and paired comparison choice experiments. *Discussion papers in economics, Dept. of Economics, University of Colorado*.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.

Vasileios Lampos, Nikolaos Aletras, Daniel Preoţiuc-Pietro, and Trevor Cohn. 2014. Predicting and characterising user impact on Twitter. In *14th Conference of the European Chapter of the Association for Computational Linguistics 2014, EACL 2014*, pages 405–413.

Tyler Lu and Craig Boutilier. 2011. Learning mallows models with pairwise preferences. In *Proceedings of the 28th international conference on machine learning (icml-11)*, pages 145–152.

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

Stephanie Lukin, Pranav Anand, Marilyn Walker, and Steve Whittaker. 2017. Argument strength is in the eye of the beholder: Audience effects in persuasion. In *15th European Chapter of the Association for Computational Linguistics (EACL)*. Association for Computational Linguistics.

Colin L Mallows. 1957. Non-null ranking models. i. *Biometrika*, 44(1/2):114–130.

Hugo Mercier and Dan Sperber. 2011. Why do humans reason? Arguments for an argumentative theory. *Behavioral and brain sciences*, 34(2):57–74.

Ariel Monteserin and Analía Amandi. 2013. A reinforcement learning approach to improve the argument selection effectiveness in argumentation-based negotiation. *Expert Systems with Applications*, 40(6):2182–2188.

Frederick Mosteller. 2006. Remarks on the method of paired comparisons: I. the least squares solution assuming equal standard deviations and equal correlations. In *Selected Papers of Frederick Mosteller*, pages 157–162. Springer.

Hannes Nickisch and Carl Edward Rasmussen. 2008. Approximations for binary Gaussian process classification. *Journal of Machine Learning Research*, 9(Oct):2035–2078.

Robin L Plackett. 1975. The analysis of permutations. *Applied Statistics*, pages 193–202.

Carl E Rasmussen and Christopher K. I. Williams. 2006. Gaussian processes for machine learning. *The MIT Press, Cambridge, MA, USA*, 38:715–719.

Steven Reece, Stephen Roberts, David Nicholson, and Chris Lloyd. 2011. Determining intent using hard/soft data and Gaussian process classifiers. In *Proceedings of the 14th International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE.

Ariel Rosenfeld and Sarit Kraus. 2016. Providing arguments in discussions on the basis of the prediction of human argumentative behavior. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 6(4):30.

Burr Settles. 2010. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11.

Edwin D Simpson, Matteo Venanzi, Steven Reece, Pushmeet Kohli, John Guiver, Stephen J Roberts, and Nicholas R Jennings. 2015. Language understanding in the wild: Combining crowdsourcing and machine learning. In *Proceedings of the 24th International Conference on World Wide Web*, pages 992–1002. International World Wide Web Conferences Steering Committee.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958.

Daniel M Steinberg and Edwin V Bonilla. 2014. Extended and unscented Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 1251–1259.

Chenhao Tan, Vlad Niculae, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. 2016. Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions. In *Proceedings of the 25th International Conference on World Wide Web*, pages 613–624. International World Wide Web Conferences Steering Committee.

Yee W Teh, Michael I Jordan, Matthew J Beal, and David M Blei. 2005. Sharing clusters among related groups: Hierarchical Dirichlet processes. In *Advances in neural information processing systems*, pages 1385–1392.

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

Ivan Titov and Alexandre Klementiev. 2012. A bayesian approach to unsupervised semantic role induction. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 12–22. Association for Computational Linguistics.

Hui Yuan Xiong, Yoseph Barash, and Brendan J Frey. 2011. Bayesian prediction of tissue-regulated splicing using RNA sequence and cellular context. *Bioinformatics*, 27(18):2554–2562.

3 Original decision letter and reviews

Starts on next page.

From: Daichi Mochihashi <daichi@ism.ac.jp>

Dear x:

As TACL action editor for submission 1304, "Finding Convincing Arguments using Scalable Bayesian Preference Learning", I am happy to tell you that I am accepting your paper subject (conditional) to your making specific revisions within two months.

LIST OF MANDATORY REVISIONS:
- Please address the common issue of the evaluation raised by multiple reviewers.
- Please prepare more explanations and experiments on scalability.
In addition to the main problems from the reviewers listed above, I would like you to add more explanations of Gaussian process preference learning itself, not just relegating it to the citations, to make the paper as self-contained as possible. This is related to our discussion over the decision on this paper: because the strength of this paper lies on the introduction of principled method of GP preference learning that could be also beneficial for other research, it should be explained well for the NLP audience at TACL.

Generally, your revised version will be handled by the same action editor (me) and the same reviewers (if necessary) in making the final decision --- which, *if* all requested revisions are made, will be final acceptance.

You are allowed one to two extra pages of content to accommodate these revisions. To submit your revised version, follow the instructions in the "Revision and Resubmission Policy for TACL Submissions" section of the Author Guidelines at https://transacl.org/ojs/index.php/tacl/about/submissions#authorGuidelines .

Thank you for submitting to TACL, and I look forward to your revised version!

Daichi Mochihashi
The Institute of Statistical Mathematics
daichi@ism.ac.jp

....THE REVIEWS....

Reviewer A:

CLARITY: For the reasonably well-prepared reader, is it clear what was done and why? Is the paper well-written and well-structured?:

4. Understandable by most readers.

ORIGINALITY/INNOVATIVENESS: How original is the approach? Does this paper
break new ground in topic, methodology, or content? How exciting and innovative is the research it describes?
Note that a paper could score high for originality even if the results do not show a convincing benefit.
:

3. Respectable: A nice research contribution that represents a notable extension of prior approaches or methodologies.

SOUNDNESS/CORRECTNESS: First, is the technical approach sound and well-chosen? Second, can one trust the claims of the paper -- are they supported by properexperiments and are the results of the experiments correctlyinterpreted?:

4. Generally solid work, although there are some aspects of the approach or
evaluation I am not sure about.

MEANINGFUL COMPARISON: Does the author make clear where the presented system
sits with respect to existing literature? Are the references adequate?:

4. Mostly solid bibliography and comparison, but there are a few additional references that should be included. Discussion of benefits and limitations is acceptable but not enlightening.

SUBSTANCE: Does this paper have enough substance (in terms of the amount of
work), or would it benefit from more ideas or analysis?:

4. Represents an appropriate amount of work for a publication in this journal. (most submissions)

IMPACT OF IDEAS OR RESULTS: How significant is the work described? If the ideas are novel, will they also be useful or inspirational? If the results are sound, are they also important? Does the paper bring new insights into the nature of the problem?:

4. Some of the ideas or results will substantially help other people's ongoing research.

REPLICABILITY: Will members of the ACL community be able to reproduce or verify the results in this paper?:

5. could easily reproduce the results.

IMPACT OF PROMISED SOFTWARE: If the authors state (in anonymous fashion)
that their software will be available, what is the expected impact of the

software package?:
3. Potentially useful: Someone might find the new software useful for their work.

IMPACT OF PROMISED DATASET(S): If the authors state (in anonymous fashion)
that datasets will be released, how valuable will they be to others?:
1. No usable datasets submitted.

TACL-WORTHY AS IS? In answering, think over all your scores above. If a paper has some weaknesses, but you really got a lot out of it, feel free to recommend it. If a paper is solid but you could live without it, let us know that you're ambivalent.

Note: after you submit this review form, you'll need to answer a related but different question via a pull-down menu: how long would it take for the authors to revise the submission to be TACL-worthy?
:
4. Worthy: A good paper that is worthy of being published in TACL.

Detailed Comments for the Authors:

The paper proposes a Gaussian process preference learning method to predict argument convincingness.

The method addresses scalability issues of previous methods, making it applicable to larger datasets. The method is shown to outperform previous methods applied for this task. In particular, it achieves significantly better results for ranking, for noisy data, and in an active learning setting.

Technically, it is a solid work, backed by an extensive set of experiments, as well as insightful error analysis. The paper is well written and easy to follow, and there is good coverage of related work.

Two main points for improvement:
(a) In the evaluation section, only the new methods make use of both linguistic and embedding features (ling+Glove), while baseline methods (SVM and BiLSTM) only use one of these sets (ling or Glove). If the authors want to clearly show that the state of the art results were achieved in part by the improved algorithm, their baselines should be the SVM and BiLSTM, each with both ling and Glove features. This is important also from a practical standpoint - SVM and BiLSTM are more accessible methods, so it should be made clear, what is the actual improvement of the new algorithm, over merely adding the Glove features to the SVM, or adding the linguistic features to the BiLSTM. This should be the baseline in any experiment that involves the SVM or the BiLSTM, unless it is shown that adding these features does not

help.

(b) While it is not possible to include complete mathematical description of the proposed method, and the authors do refer to the relevant related work, it would help to provide some more details about the learning method in Section 5. For instance, what q is, and how it is updated.

Minor comments:
Section 3, the paragraph the follows equation (1): which learns the the number of topics than --> rephrase

References: please review your references, make them consistent, fix capitalization etc. Also, fix Reece et al: In proceedings of the ... conference on (*), pages ...

REVIEWER CONFIDENCE:
4. Quite sure. I tried to check the important points carefully. It's unlikely, though conceivable, that I missed something that should affect my ratings.

Reviewer B:

CLARITY: For the reasonably well-prepared reader, is it clear what was done and why? Is the paper well-written and well-structured?:
4. Understandable by most readers.

ORIGINALITY/INNOVATIVENESS: How original is the approach? Does this paper
break new ground in topic, methodology, or content? How exciting and innovative is the research it describes?
Note that a paper could score high for originality even if the results do not show a convincing benefit.
:
3. Respectable: A nice research contribution that represents a notable extension of prior approaches or methodologies.

SOUNDNESS/CORRECTNESS: First, is the technical approach sound and well-chosen? Second, can one trust the claims of the paper -- are they supported by properexperiments and are the results of the experiments correctlyinterpreted?:
2. Troublesome. There are some ideas worth salvaging here, but the work should really have been done or evaluated differently.

MEANINGFUL COMPARISON: Does the author make clear where the presented

system
sits with respect to existing literature? Are the references adequate?:
2. Only partial awareness and understanding of related work, or a flawed comparison or deficient comparison with other work.

SUBSTANCE: Does this paper have enough substance (in terms of the amount of
work), or would it benefit from more ideas or analysis?:
3. Leaves open one or two natural questions that should have been pursued
within the paper.

IMPACT OF IDEAS OR RESULTS: How significant is the work described? If the ideas are novel, will they also be useful or inspirational? If the results are sound, are they also important? Does the paper bring new insights into the nature of the problem?:
3. Interesting but not too influential. The work will be cited, but mainly for comparison or as a source of minor contributions.

REPLICABILITY: Will members of the ACL community be able to reproduce or verify the results in this paper?:
4. could mostly reproduce the results, but there may be some variation because of sample variance or minor variations in their interpretation of the protocol or method.

IMPACT OF PROMISED SOFTWARE: If the authors state (in anonymous fashion)
that their software will be available, what is the expected impact of the software package?:
3. Potentially useful: Someone might find the new software useful for their work.

IMPACT OF PROMISED DATASET(S): If the authors state (in anonymous fashion)
that datasets will be released, how valuable will they be to others?:
1. No usable datasets submitted.

TACL-WORTHY AS IS? In answering, think over all your scores above. If a paper has some weaknesses, but you really got a lot out of it, feel free to recommend it. If a paper is solid but you could live without it, let us know that you're ambivalent.

Note: after you submit this review form, you'll need to answer a related but different question via a pull-down menu: how long would it take for the authors to revise the submission to be TACL-worthy?
:
2. Leaning against: I'd rather not see it appear in TACL.

Detailed Comments for the Authors:

This paper introduces a scalable Bayesian preference learning method for identifying convincing arguments. The most interesting thing about this method is that it does not require separate methods to produce training data, predict rankings and perform pairwise classifications. When applied to identifying convincing arguments, the method learns from training data composed of pairwise judgments a model for predicting a convincingness score for each argument, from which a ranking of the arguments can be derived. I also find Experiment 1 and the experiment on active learning interesting. Experiment 1 helps the reader understand how the models differ from each other, while the active learning experiment provides suggestive evidence that the model can do a better job than other classifiers for selecting the best instance to annotate.

Overall, the paper is clearly written and is fairly easy to follow. I do think, however, that the major concerns listed below need to be addressed before this paper can be recommended for publication in TACL.

1) Confusing statements

Abstract: "We introduce a scalable Bayesian preference learnng method for identifying convincing arguments in the absence of gold-standard ratings or rankings." This sentence led me to think that this method is unsupervised. Your method does need gold-standard (pairwise) rankings.

In the intro and throughout the paper, you keep emphasizing that the pairwise preferences are noisy. While I understand that the annotations may be noisy because they were obtained by means of crowdsourcing, I don't understand the role played by the noise in the learning process, particularly since there were no results on noise-free data for comparison purposes. In addition, you believe they are noisy, then should we trust the results because they were also evaluated on noisy test data?

2) The major claim lacks substantiation

The main claim is that the proposed preference learning method is scalable, but there isn't any experiment on illustrating how scalable the method is. Section 5 talks about using M inducing points. What value of M did you use? How did you decide that this value should be used? How scalable is this method w.r.t. different values of M? And how sensitive is the method to different values of M?

3) Evaluation issues

I don't understand why the SVM and the BiLSTM were not trained on

"ling+Glove", especially since using both types of features seem to offer better performance. If I understand correctly, the claim is that your model performs better than existing classifiers (LSTM and SVM). The claim is *not* that your model is better than existing classifiers after given a better a feature set. So a fair comparison would involve giving all models the same feature set. In other words, I don't think the comparison in Table 3 is fair at all.

There is no discussion of how the parameters of the SVM and the BiLSTM are set, so it's not clear whether the performance of these classifiers suffered because their parameters were not properly tuned.

It isn't fair to use the SVM/BiLSTM regression results as baselines for the ranking experiments. In fact, it's strange that you needed to cast the SVM/BiLSTM ranking experiments as regression problems and then convert the regression results to ranking results. Many advanced ranker learning algorithms have been developed in the machine learning community in recent years, so why not train them on the PageRank output to directly obtain the ranking?

In Section 6.2, you set the hyperparameters a0 and b0 to certain values "after testing three different settings". Which three settings did you do? Were they obtained by cross validation on the training data? What was the criterion used in selecting these parameters among the three settings?

Overall, it doesn't seem ideal to do ranking experiments on this dataset - as you said, the rankings were automatically derived using PageRank from the pairwise classifications, and in addition, your discussion in Experiment 1 seemed to suggest that PageRank doesn't always produce intuitive rankings. So it seems strange that on one hand, you suggest that the rank produced by your model is better than that of PageRank and on the other hand you use the PageRank rankings as gold standard and evaluate your model's rankings against the PageRank rankings. Since the method (and the resulting model) isn't specifically designed for determining argument convincingness, perhaps it'd be better to evaluate it on tasks where gold-standard rankings are available.

Given these issues, I don't think the paper presented convincing empirical evidence that the proposed model is better than the baselines.

4) Insights into argument convincingness

I am not sure what I learned about the argument convincingness problem from this paper. While I understand that the model seems to produce better results than the baselines, I don't know what linguistic insights I gained into the argument convincingness problem other than the fact that some features were preferred by certain models from Section 6.8. Although the

focus of this paper is the model, without a proper discussion of the linguistic insights that the model provides, the contribution of this paper will merely be an application of an (improved) machine learning technique to a (pairwise) ranking problem with better results.

5) related work

Besides Habernal and Gurevych (2016), there are other papers on automatically determining argument convincingness (e.g.,Liu et al. (ArgMin workshop 2016), Persing & Ng (IJCAI 2017)). Some of them employ different schemes for annotating convincingness than Habernal and Gurevych's and should be discussed. I suggest that the authors did a more thorough investigation of related work.

REVIEWER CONFIDENCE:

4. Quite sure. I tried to check the important points carefully. It's unlikely, though conceivable, that I missed something that should affect my ratings.

Reviewer C:

CLARITY: For the reasonably well-prepared reader, is it clear what was done and why? Is the paper well-written and well-structured?:

4. Understandable by most readers.

ORIGINALITY/INNOVATIVENESS: How original is the approach? Does this paper
break new ground in topic, methodology, or content? How exciting and innovative is the research it describes?
Note that a paper could score high for originality even if the results do not show a convincing benefit.
:

3. Respectable: A nice research contribution that represents a notable extension of prior approaches or methodologies.

SOUNDNESS/CORRECTNESS: First, is the technical approach sound and well-chosen? Second, can one trust the claims of the paper -- are they supported by properexperiments and are the results of the experiments correctlyinterpreted?:

4. Generally solid work, although there are some aspects of the approach or
evaluation I am not sure about.

MEANINGFUL COMPARISON: Does the author make clear where the presented

system
sits with respect to existing literature? Are the references adequate?:
4. Mostly solid bibliography and comparison, but there are a few additional references that should be included. Discussion of benefits and limitations is acceptable but not enlightening.

SUBSTANCE: Does this paper have enough substance (in terms of the amount of
work), or would it benefit from more ideas or analysis?:
4. Represents an appropriate amount of work for a publication in this journal. (most submissions)

IMPACT OF IDEAS OR RESULTS: How significant is the work described? If the ideas are novel, will they also be useful or inspirational? If the results are sound, are they also important? Does the paper bring new insights into the nature of the problem?:
3. Interesting but not too influential. The work will be cited, but mainly for comparison or as a source of minor contributions.

REPLICABILITY: Will members of the ACL community be able to reproduce or verify the results in this paper?:
4. could mostly reproduce the results, but there may be some variation because of sample variance or minor variations in their interpretation of the protocol or method.

IMPACT OF PROMISED SOFTWARE: If the authors state (in anonymous fashion)
that their software will be available, what is the expected impact of the software package?:
1. No usable software released.

IMPACT OF PROMISED DATASET(S): If the authors state (in anonymous fashion)
that datasets will be released, how valuable will they be to others?:
1. No usable datasets submitted.

TACL-WORTHY AS IS? In answering, think over all your scores above. If a paper has some weaknesses, but you really got a lot out of it, feel free to recommend it. If a paper is solid but you could live without it, let us know that you're ambivalent.

Note: after you submit this review form, you'll need to answer a related but different question via a pull-down menu: how long would it take for the authors to revise the submission to be TACL-worthy?
:
4. Worthy: A good paper that is worthy of being published in TACL.

Detailed Comments for the Authors:

This paper presents a method for identifying convincing arguments based on Bayesian methods including Preference learning and Gaussian Process classification. The paper uses the publicly available data set developed in (Habernal & Gurevych, ACL 2016), which has two tasks: pairwise classification and argument ranking within a topic. Both tasks are attempted with the adjudicated labels experimented in the original paper and with noisy and possibly conflicting labels. The results show that the current approaches obtain consistent improvements.

The paper is overall well-written and well-structured. The problem is well suited to the techniques proposed. In particular, Gaussian Process Preference Learning is highlighted, a technique which has not been used in NLP, although previous research showed that other variations of Gaussian Processes lead to good results in a series of NLP tasks. The paper applies a stochastic variational inference approach inspired by previous work on Gaussian Processes to handle the size of the data set. Personally, I am surprised that the method is able to scale to use all the linguistic features (>30000), as this is the main limitation of using Gaussian Processes. To this end, I would be interested in a more complete rundown of running times (in addition to the couple of numbers already presented in page 8).

The methods compared against include the two original implementations described in (Habernal & Gurevych, ACL 2016) and the same experimental setup (the results from this paper are better in some cases than the original results, maybe worth explaining why). In addition, a couple of other related methods are compared against: preference learning in the SVM framework as a non-bayesian alternative and Gaussian Process classification as a more conventional non-linear classification alternative. However, I would like to see the comparison being made with SVM's using the same features (ling + Glove) as the new methods. The results show that overall Gaussian Process classification performs best for pairwise classification and GPPL performs best for ranking. The first consideration should be taken into account when making the claims in the abstract and introduction as the classic Gaussian Process performs generally better than the preference learning. I think that, similar to the ULPConvArgStrict/Rank, the tasks in Table 3 should also be split into classification and ranking when presented.

The active learning experiments are a nice addition to the paper. However, it would be preferable to see in Figure 6 the graphs for the entire set of pairs beyond 200.

I find the interpretation section to not be at the level of the rest of the paper. Learning of the ARD lengthscales is one of the key advantages of using Gaussian Processes that facilitates interpretability. I would prefer

to see a more detailed analysis of the top features (lowest lengthscales) in each class of features and any interpretation of why these are useful for predicting convincing arguments. This would also help with the error analysis which is a nice addition to the paper but lacks grounding in the experimental results. Some assumptions are made in that section about convincing/unconvincing arguments (language errors, all caps sentences, multiple exclamation marks, not strongly taking a side) which should have been derived from the data, rather than anecdotical. Also, for this goal Glove embeddings are not useful as they are not interpretable (e.g. what does embedding dimension 19 represent?). For something similar, see the approach of (Preotiuc-Pietro, Lampos, Aletras ACL 2015) where they use word embedding derived topics.

I am not sure of the validity of the claim 'the embeddings are only weakly informative', as table 2 shows that these are highly predictive of the outcome on their own and improve performance when added on top of the other linguistic features. We can also observe that some have very low lengthscales, although the average and median are high for the feature class. I believe this is due to their high coverage when compared to e.g. unigrams - the latter are very sparse, while the embeddings are non-zero for each example.

If space is a concern, I personally did not find the toy data sets to contribute much to the paper + the overview of the paper at the end of the intro can be removed.

I think that the abstract, introduction and conclusion should be toned down and more accurately reflect the findings in the paper:
- in the abstract: "in contrast to previous, work, we avoid the need for separate methods to produce training data, predict rankings and perform pairwise classification" - I do not understand this claim: Training data is obtained from previous work, and previous methods were the same for ranking and classification (whether LSTM or SVM)
- the introduction takes quite long to get to the contributions of this paper. I think some of the initial material which is not novel of this work should be moved to Section 2.
- in the conclusion: "We showed that the method performs well with noisy training data, reducing dependence on a quality control pipeline for crowdsourced data." - I do not find this to be true from the experiments as performance on the second setup is still lower and the patterns of improvement are similar to that of SVMs and LSTMs.

Other considerations:
- I do not think the example in Figure 1 is well chosen as it is quite specific and context is missing (e.g. the ratings for each argument)
- why is the Matern 3/2 kernel used? Were any other kernels tried e.g. the RBF or Matern 5/2?

- what is the motivation for using the inverse scale gamma prior? (Section 5, first paragraph).
- did you try other methods for selecting the inducing points? How about random selection rather than K-means?
- Table 2: move 'UKPConvArgStrict' two rows down.
- ROC AUC different is not measured in percent (Section 6.4), but in absolute 0 to 1 score.
- 'highly statistically significant' - improper use of 'highly', just statistically significant and threshold + the test performed is enough.
- 'rape' is not an emotion term.
- why is 'safer' the most predictive unigram feature, while 'safest' and 'safety' the least?

REVIEWER CONFIDENCE:

4. Quite sure. I tried to check the important points carefully. It's unlikely, though conceivable, that I missed something that should affect my ratings.
