Modelling Uncertainty in Collaborative Document Quality Assessment

Aili Shen **Daniel Beck** **Bahar Salehi** **Jianzhong Qi** **Timothy Baldwin**
School of Computing and Information Systems
The University of Melbourne
Victoria, Australia
ailis@student.unimelb.edu.au
d.beck@unimelb.edu.au baharsalehi@gmail.com
jianzhong.qi@unimelb.edu.au tb@ldwin.net

Abstract

In the context of document quality assessment, previous work has mainly focused on predicting the quality of a document relative to a putative gold standard, without paying attention to the subjectivity of this task. To imitate people's disagreement over inherently subjective tasks such as rating the quality of a Wikipedia article, a document quality assessment system should provide not only a prediction of the article quality but also the uncertainty over its predictions. This motivates us to measure the uncertainty in document quality predictions, in addition to making the label prediction. Experimental results show that both *Gaussian processes* (GPs) and *random forests* (RFs) can yield competitive results in predicting the quality of Wikipedia articles, while providing an estimate of uncertainty when there is inconsistency in the quality labels from the Wikipedia contributors. We additionally evaluate our methods in the context of a semi-automated document quality class assignment decision-making process, where there is asymmetric risk associated with overestimates and underestimates of document quality. Our experiments suggest that GPs provide more reliable estimates in this context.

1 Introduction

The volume of textual web content generated collaboratively — through sites such as Wikipedia, or community question answering platforms such as Stack Overflow — has been growing progressively. Such collaborative paradigms give rise to a problem in quality assessment: how to ensure documents are reliable and useful to end users.

Given the volume of such documents, and velocity with which they are being produced, there has been recent interest in *automatic* quality assessment using machine learning techniques (Dang and Ignat, 2016a; Dalip et al., 2017; Shen et al., 2017). However, previous work has treated this problem using off-the-shelf predictors, which fail to take into account two key aspects. First, any quality rating is inherently *subjective*: different end users can heavily disagree on the quality of a document. For example, as shown in Figure 1, the Wikipedia article *Warden Head Light*[1] is assigned to different labels from different Wikipedia Projects:[2] B (in the green block) by Wikipedia Project *Lighthouses*, and Start (in the orange block) by Wikipedia Project *Australia*;[3] among a 30K datatset we collected, there are 7% such articles (even including high-quality articles), where contributors disagree over the article quality. Second, previous work has ignored *decision-making*

Figure 1: A screenshot of the "Warden Head Light" Talk page. Wikipedia Project *Lighthouses* assigns a B-class quality label to this article, while Wikipedia Project *Australia* assigns a Start-class quality label.

[1] https://en.wikipedia.org/w/index.php?title=Warden_Head_Light&oldid=759074867

[2] A Wikipedia Project is a group of Wikipedia contributors who work together to improve Wikipedia articles that they are interested in.

[3] We return to describe the full label set in Section 2.

Proceedings of the 2019 EMNLP Workshop W-NUT: The 5th Workshop on Noisy User-generated Text, pages 191–201
Hong Kong, Nov 4, 2019.

procedures (such as expert reviewing, and featuring articles on the Wikipedia main page) that are impacted by the results of the prediction, which can vary in non-trivial ways.

In this work, we address these two gaps by modelling the *uncertainty* in the quality labels by treating predictions as probability distributions. In order to obtain these distributions, we experiment with both Bayesian models (Gaussian Processes, GPs, Rasmussen and Williams, 2006) and frequentist, ensemble-based methods (Random Forests, RFs, Breiman, 2001), applying them to English Wikipedia articles. Our results show that these approaches are competitive with the state-of-the-art in terms of predictive performance, while also providing estimates of uncertainty in the form of predictive distributions.

As a case study on the utility of uncertainty estimates, we analyse a typical Wikipedia scenario, where articles with predicted high quality are sent to expert reviewers to confirm their status. Such reviewing procedures are costly: if a low-quality article is predicted to be a featured article (the highest quality in Wikipedia), the triggered manual review can substantially waste time and human effort. Conversely, if a high-quality article is predicted to be of a lower-quality class, there is no cost to the editor community.[4] This is an example of asymmetric risk, where underestimates and overestimates have different penalties. In this paper, we show how to use uncertainty estimates from predictions in order to make a quality prediction that minimises this asymmetric risk.

In summary, this paper makes the following contributions:

(i) We are the first to propose to measure the uncertainty of article quality assessment systems. We find that both GPs and RFs can achieve performance competitive with the state-of-the-art, while providing uncertainty estimates over their predictions in the form of predictive distributions.

(ii) To model asymmetric risk scenarios in Wikipedia, we propose to combine the predictive distributions provided by our methods with asymmetric cost functions. Experimental results show that GPs are superior to RFs under such scenarios.

(iii) We constructed a 30K Wikipedia article dataset containing both gold-standard labels and Wikipedia Project labels, which we release for public use along with all code associated with this paper at https://github.com/AiliAili/measure_uncertainty.

[4] Although there may be an opportunity cost (in terms of not showcasing high-quality articles), and the potential demotivation of the associated editors.

2 Preliminaries

In this section, we detail the specific scenario addressed in this study: quality assessment of Wikipedia articles. We also describe the procedure to construct our dataset.

2.1 Problem Definition

In line with previous work (Warncke-Wang et al., 2015; Dang and Ignat, 2016a,b, 2017), we consider six quality classes of Wikipedia articles, ordered from highest to lowest: Featured Article ("FA"), Good Article ("GA"), B-class Article ("B"), C-class Article ("C"), Start Article ("Start"), and Stub Article ("Stub"). A description of the quality grading criteria can be found in the Wikipedia grading scheme page.[5]

The quality assessment process over a Wikipedia article is done in a collaborative way, through discussions on the corresponding article's Talk page.[6] Wikipedia contributors also carefully review articles that are GA and FA candidates. In particular, FA articles are eligible to appear on the main page of the website. A reliable automatic quality assessment model should take these decision making aspects into account.

Problem statement. In this paper, our aim is to predict the quality of unseen Wikipedia articles, paired with an estimate of uncertainty over each prediction, which we evaluate in a risk-aware decision making scenario. Figure 2 summarises model application and actions depending on the uncertainty: (1) quality-indicative features are first extracted from a Wikipedia article; (2) a model predicts the article quality and provides an indication of how confident it is of its prediction; and (3) different actions are taken based on the predicted quality and confidence value, such as expert review and featuring on the Wikipedia main page if an article is predicted to be FA/GA with high confidence.

[5] https://en.wikipedia.org/wiki/Template:Grading_scheme

[6] Such as the Talk page for the "Warden Head Light" article: https://en.wikipedia.org/wiki/Talk:Warden_Head_Light

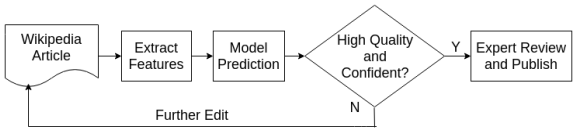

Figure 2: Model application and decision-making procedure.

2.2 Data Collection

We constructed an evaluation dataset by collecting articles from Wikipedia in a balanced way. From each quality class, we crawled 5K articles from its corresponding repository.[7] As mentioned in Section 1, quality assessment is subjective and multiple editors/Wikipedia Projects may disagree when assigning a quality label to an article. We can observe this behaviour by inspecting an article's corresponding Talk page, which records quality labels from different Wikipedia Projects.[8] For roughly 7% of the articles, there is a disagreement between editors/Wikipedia Projects. Take Figure 1 in Section 1 as an example, although the *primary* label of the Wikipedia article *Warden Head Light* is B, two other quality labels are assigned to it: B class (in the green block) by Wikipedia Project *Lighthouses*, and Start class (in the orange block) by Wikipedia Project *Australia*. Since we are interested in investigating how an automatic quality assessment system performs when there is a disagreement, we also crawl these *secondary* labels when building our dataset. Finally, we remove markup that relates to the document quality classes, such as {*Featured Article*} or {*geo-start*}, to alleviate any overt indication of the quality label in the text of the article.

The resulting dataset contains 29,097 Wikipedia articles, which we partition into two subsets for separate evaluation in Section 4: (1) *consistent* articles, where *primary* and *secondary* labels fully agree; and (2) *inconsistent* articles, where there is disagreement among *secondary* labels, with at least one of them agreeing with the *primary* label. We emphasise that we keep *all secondary* labels for the latter, without performing any label aggregation (e.g., voting). Our aim is to make quality predictions as close as possible to the *primary* labels while also providing uncertainty estimates of such predictions: lower uncertainty over *consistent* articles and higher uncertainty over *inconsistent* articles. The dataset is then stratified into training, development, and test sets, as detailed in Table 1.

		Train	Dev	Test	Total
FA	consistent	3956	470	538	4998
	inconsistent	28	4	2	
GA	consistent	3887	468	495	4878
	inconsistent	16	6	6	
B	consistent	3138	400	416	4843
	inconsistent	702	85	102	
C	consistent	3036	382	381	4523
	inconsistent	570	69	85	
Start	consistent	3725	451	472	4924
	inconsistent	223	28	25	
Stub	consistent	3863	470	492	4931
	inconsistent	83	12	11	
Total	—	23227	2845	3025	29097

Table 1: A breakdown of our Wikipedia dataset.

3 Methods

A key aspect of the task is the ordinal nature of the quality labels, e.g., a Start article is close in quality to a C, but much worse than an FA. Surprisingly though, most previous studies (Dang and Ignat, 2016a,b, 2017; Shen et al., 2017) formulate the problem as multi-class classification and use accuracy as the evaluation metric. Such modelling and evaluation procedures completely disregard the ordinal nature of the labels, which in turn does not correspond to real world scenarios: the cost of mispredicting a Start article as C is different to mispredicting it as an FA (standard classification metrics such as accuracy assume equal cost for all mispredictions).

To better address the scenarios we are interested in, we treat quality assessment as a regression problem, in terms of both modelling and evaluation. In order to do this, we encode the quality class labels as real values by mapping them to the interval $[-2.5, 2.5]$ with increments of 1. These labels are -2.5, -1.5, -0.5, 0.5, 1.5, and 2.5, respectively, where higher values indicate higher quality.[9] We perform this step to be able to use off-the-shelf regression models, while also center-

[7] For example, we obtain FA articles by crawling pages from the FA repository: https://en.wikipedia.org/wiki/Category:Featured_articles

[8] Different Wikipedia articles can be rated by different Wikipedia Projects. And the number of quality labels in a Talk page depends on how many Wikipedia Projects rate this article.

[9] Having equal intervals is a heuristic: we discuss this limitation in Section 6.

ing the labels. The remainder of this section details the regression methods we use, as well as two types of features we employ to represent each article. Both methods provide uncertainty estimates through *predictive distributions* (Gaussian distributions in our case).

3.1 Gaussian Processes

A principled approach to obtain predictive distributions is to use GPs (Rasmussen and Williams, 2006), a Bayesian non-parametric framework widely considered the state-of-the-art for regression (Hensman et al., 2013). Given a latent function f, which explains the relationship between an input vector $\boldsymbol{x}$ and its corresponding output value y, the model assumes that f is distributed according to a GP, i.e.,

$$f(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x'})),$$

where $m(\boldsymbol{x})$ is a mean function, and $k(\boldsymbol{x}, \boldsymbol{x'})$ is a covariance or *kernel* function.

Following common practice, we fix the mean function to zero, as our output values are centered. Most of the information obtained from the training data can be encoded in the kernel function, of which the most common one is the Radial Basis Function (RBF), defined as

$$k(\boldsymbol{x}, \boldsymbol{x'}) = \sigma_v \exp\left(-\frac{1}{2}\sum_{j=1}^{d}\frac{1}{\ell_j^2}(x_j - x'_j)^2\right),$$

where σ_v is the variance hyperparameter controlling the scale of the labels, and ℓ_j is the lengthscale for the jth dimension of the input. The lengthscales are learned by maximising the marginal likelihood (Rasmussen and Williams, 2006), resulting in a feature selection procedure known as *Automatic Relevance Determination* (ARD): lower lengthscales indicate features with higher discriminative power. We use this procedure to perform a feature analysis in Section 4.1. Besides the RBF, we also experiment with a range of other kernels used in GP models: see Rasmussen and Williams (2006, Chap. 4) for details.

Standard GP inference takes $\mathcal{O}(n^3)$ time, where n is the number of instances in the training data. As this is prohibitively expensive given the size of our dataset, we employ Sparse GPs (Titsias, 2009; Gal et al., 2014), a scalable extension that approximates an exact GP by using a small set of latent *inducing points*. These are learned by maximising a variational lower bound on the marginal likelihood: see Titsias (2009) for details.

3.2 Random Forests

As an alternative method to obtain predictive distributions, we use RFs (Breiman, 2001), which are ensembles of decision trees (regression trees in our case). Each tree is trained on a bootstrapped sample of the training set and within each tree, a random subset of features is used when splitting nodes. To obtain predictive distributions, we assume that the individual tree predictions follow an "empirical" Gaussian distribution. The mean and the variance are computed from the full set of predictions obtained by the RF. While this approach is less principled — since there is no reason to believe the distribution over individual predicted values is Gaussian — it can work well in practice. For instance, RFs have been used before to obtain uncertainty estimates in the context of Bayesian Optimisation (Hutter et al., 2011).

3.3 Features and Preprocessing

Following Dang and Ignat (2016a), we use hand-crafted features, in the form of 11 structural features and 10 readability scores, which are listed in Dang and Ignat (2016a) and Shen et al. (2017).[10] Structural features can reflect the quality of Wikipedia articles in different ways. For example, *References*, *Pagelinks*, and *Citation* show how the article content is supported by information from different sources, indicating whether the article is reliable and thus indicating higher/lower quality, while features *Level2*, and *Level3+* indicate how the content is organised, which is another quality indicator of Wikipedia articles. Readability scores reflect the usage of language and comprehension difficulty of a Wikipedia article. For example, *Difficult Words* (Chall and Dale, 1995), *Dale-Chall* (Dale and Chall, 1948), and *Gunning-Fog* (Gunning, 1969) use the number or percentage of difficult words to measure the comprehension difficulty of a text, where a difficult word is a word not in a list of predefined words that fourth-grade American students can reliably understand. These hand-crafted features are extracted from Wikipedia articles using the open-source packages

[10] Dang and Ignat (2016a) explore nine readability scores, to which we add an extra readability score denoted *Consensus*. This score represents the estimated school grade level required to understand the content.

wikiclass[11] and *textstat*.[12]

As features from the revision history, such as the number of revisions and the article–editor network, are indirect quality indicators, we only focus on direct quality indicators from the content itself.

4 Experimental Study

In this section, we detail four sets of experiments: (1) intrinsic comparison of our methods with respect to their predictive distributions; (2) comparative experiments with the state-of-the-art with respect to point estimates only; (3) experiments measuring the performance of our methods in a transfer setting, where the goal is to predict *secondary* labels; and (4) a case study where automatic labels are used to filter articles for manual revisions and we use distributions to incorporate risk in the quality predictions. All our models are trained on the *primary* labels but we explicitly report our results on two different test sets: one with *consistent* and one with *inconsistent* labels, as explained in Section 2.2.

4.1 Intrinsic Evaluation

Our first set of experiments evaluates the performance of methods intrinsically, with respect to their predictive distributions.

GP settings. We use GP models from the *GPflow* toolkit (Matthews et al., 2017). In particular, we use a Sparse GP with 300 inducing points, which are initialised with k-means clusters learned on the training set and we explore different kernels (RBF, Arccosine (Cho and Saul, 2009), Matérn 32, Matérn 52, Rational Quadratic (RQ)).

RF settings. We use the RF implementation in *scikit-learn* (Pedregosa et al., 2011), with 300 trees and a maximum depth of 40, fine-tuning over the development set. All other hyperparameters are set to default values.

Evaluation metrics. Standard metrics to evaluate regression models such as Root Mean Squared Error (RMSE) and Pearson's correlation (r) are only based on *point estimate* predictions. These are not ideal for our setting since we aim to assess predictive *distributions* instead. For such settings, Candela et al. (2005) proposed the Negative Log Predictive Density ("NLPD") as an alternative metric, which is commonly used in the literature (Chalupka et al., 2013; Hernández-Lobato and Adams, 2015; Beck et al., 2016) to evaluate probabilistic regression models.

Given a test set containing the input and its reference score $(\boldsymbol{x}_i, y_i)$, NLPD is defined as

$$\text{NLPD} = -\frac{1}{n}\sum_{i=1}^{n} \log p(\hat{y}_i = y_i | \boldsymbol{x}_i),$$

where n is the number of test samples and $p(\hat{y}_i|\boldsymbol{x}_i)$ is the predictive distribution for input $\boldsymbol{x}_i$. For Gaussian distributions, NLPD penalises both overconfident wrong predictions and underconfident correct predictions.

	NLPD (*consistent*)	NLPD (*inconsistent*)
RF	**0.978**†	1.642
GP_{RBF}	1.224	**1.364**†
GP_{arc0}	1.280	1.460
GP_{arc1}	1.266	1.428
GP_{arc2}	1.286	1.426
$\text{GP}_{\text{Matérn32}}$	1.275	1.427
$\text{GP}_{\text{Matérn52}}$	1.275	1.425
GP_{RQ}	1.271	1.442

Table 2: Intrinsic evaluation results. GP_{arc0}, GP_{arc1}, and GP_{arc2} denote GP using an Arccosine kernel with orders of 0, 1, and 2. $\text{GP}_{\text{Matérn32}}$, $\text{GP}_{\text{Matérn52}}$, and GP_{RQ} denote GP using Matérn32, Matérn52, and RQ, respectively. The best result is indicated in **bold**, and marked with "†" if the improvement is statistically significant (based on a one-tailed Wilcoxon signed-rank test; $p < 0.05$).

Results. Table 2 shows the average NLPD over 10 runs on both test sets. Clearly, RFs outperform GPs on the consistent set while the opposite happens on the inconsistent set. In general, this shows that GPs tend to give more conservative predictive distributions compared to RFs. This is beneficial when there is label disagreement, and therefore, high uncertainty over the labels. However, it also translates into worse performance when labels are consistent, where the higher confidence obtained by RFs give better results. In terms of kernels, we obtained the best results using RBF for both test sets.

Feature Analysis. To find out which features contribute most to the performance of our models, we analyse the lengthscales from GP and the feature weigths from RF. Figure 3 shows the top

[11] https://github.com/wiki-ai/wikiclass

[12] https://pypi.python.org/pypi/textstat/0.5.1

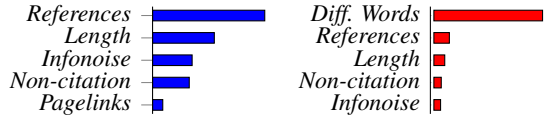

Figure 3: Feature importance values in the GP model (left) and RF model (right) . A full description of all features can be found in Dang and Ignat (2016a) and Shen et al. (2017).

five features in each model: four of those shared by both GP and RF, indicating their importance as quality indicators in both methods. In particular, the length of an article is a consistently good indicator: a possible explanation is that short articles lack enough content to be considered high quality articles. The presence of structured indicators such as number of references and non-citation templates is also interesting, as it is evidence of the contribution of non-textual information in the quality assessment process.

4.2 Point Estimate Comparison

In terms of point estimates, the state-of-the-art in our task is a neural model based on a BILSTM architecture (Shen et al., 2017). As we model quality prediction of Wikipedia articles as a regression problem, where a linear transformation is used as the final layer instead of a softmax, the neural network model does not provide predictive distributions, limiting their applicability in the scenarios which we focus on in this work. However, to put our proposed methods into perspective, we compare their performance with these neural models in terms of point estimates. Specifically, we use the mean of the distributions for both GP and RF as predictions and use standard regression metrics (RMSE and Pearson's r correlation) to assess the performance of our models against BILSTM.

For the GP model, we restrict our evaluation in this section (and in the remainder of this paper) to the one with an RBF kernel, as it performed significantly better in Section 4.1. We compare with two BILSTM models: (1) with pre-trained word embeddings, using GloVe (Pennington et al., 2014) ("BILSTM$^+$" hereafter); and with randomly initialised word embeddings ("BILSTM$^-$" hereafter). See Shen et al. (2017) for a detailed description of all hyperparameters.

Results. From Table 3, we see that while BILSTM$^+$ outperform our methods in the *consistent* set, the difference is small and we obtain good results nevertheless. In particular, correlation is close to 0.9 for all methods. Therefore, we can see that GPs and RFs obtain comparable results with the state-of-the-art while providing additional information through the predictive distributions.

	consistent		*inconsistent*	
	RMSE	r	RMSE	r
BILSTM$^+$	**0.795**†	**0.897**†	0.951	0.522
BILSTM$^-$	0.810	0.891	0.936	**0.548**†
RF	0.805	0.892	0.942	0.527
GP_{RBF}	0.822	0.887	**0.932**	0.545

Table 3: Point estimate comparison results.

The importance of having distributions as predictions becomes clear when we see the results for the *inconsistent* set, in Table 3. Here, not only do GPs perform on par with the BILSTM models, but the overall correlation is much lower (between 0.52 and 0.55). This highlights the harder task of predicting quality labels under disagreement, which further motivates the additional uncertainty information coming from predictive distributions.

4.3 Prediction of Secondary Labels

As explained in Section 2.2, the *inconsistent* articles are ones where the *primary* label is in disagreement with the *secondary* ones, from different Wikipedia Projects. In this section, we assess how our models fare under a transfer scenario, where the goal is to predict these secondary labels. Such a scenario can be useful, for instance, if we want to incorporate information from Projects to decide the quality of a document.

To measure the performance with *secondary* labels as references, one option is to aggregate the labels of an article into a single one (through voting or averaging, for instance) and use that value as the reference. Instead, we opt to embrace the disagreement, and propose a *weighted* extension of NLPD, namely wNLPD, which we define as

$$\text{wNLPD} = -\frac{1}{n}\sum_{i=1}^{n}\log\sum_{j=1}^{m_i} w_j p(\hat{y}_i = y_j|\boldsymbol{x_i}),$$

where m_i is the number of *secondary* labels for article $\boldsymbol{x}_i$, and w_j is the weight for label j. If we have prior information about the reliability of some label sources (for instance, different

	wNLPD	wRMSE
RF	1.460	1.158
GP_{RBF}	**1.412**†	**1.152**†

Table 4: Results for prediction of secondary labels.

Wikipedia Projects), one can plug this information into the weights. Here we assume equal reliability and use uniform weights $w_j = \text{freq}(j)/m_i$, where freq is the count of label j among *secondary* labels. The metric degrades to standard NLPD when labels are consistent. We also evaluate point estimate performance using a similar weighting scheme for RMSE (which we denote as wRMSE).

Results. Table 4 summarises the results. Notice that in this setting we only report results for the *inconsistent* test set, as the *consistent* one has no disagreement (and therefore, numbers would match the ones in Section 4.1). Here we also see that GPs achieve significantly better performance than RFs, although by a much lower margin compared to the results on the *primary* labels (Table 2). This reflects the harder aspect of this setting. We hypothesize we can obtain better performance in this scenario by incorporating the *secondary* labels at training time, which we leave for future work.

4.4 Case Study: Quality Prediction as Filtering for Manual Revision

As mentioned in Section 1, one use for a quality prediction system is to filter documents for manual revision. In the case of Wikipedia, such revisions are mandatory for articles to be assigned as a Good Article or a Featured Article. This incurs in an *asymmetric risk*: the cost of mispredicting an article as GA and FA is higher than other labels, as these trigger expensive, manual labour. Such a scenario can be modelled through *asymmetric loss functions* (Varian, 1975).

If a quality model provides predictive distributions, one can obtain *optimal* quality decisions under an asymmetric loss function through the framework of Minimum Bayes Risk (MBR). This setting has been studied before by Christoffersen and Diebold (1997) and more recently applied by Beck et al. (2016) in the context of machine translation post-editing. However, these assume a regression scenario. While we employ regression models in our work for ease of modelling reasons, the final decisions in the pipeline are discrete (although still ordinal).

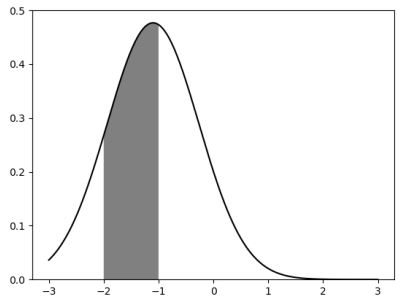

Figure 4: Discretisation of a continuous predictive distribution. The shaded area shows the probability of quality label Start (-1.5).

To adapt the MBR framework into our case, we first define the risk $\delta(q)$ of predicting the quality label q as

$$\delta(q) = \sum_{\hat{q}} \mathcal{L}(\hat{q}, q) p(\hat{q}|\boldsymbol{x}),$$

where $\mathcal{L}(\hat{q}, q)$ is an (asymmetric) loss function over the discrete quality labels and $p(\hat{q}|\boldsymbol{x})$ is the *discretised* probability of quality label $\hat{q}$ for document $\boldsymbol{x}$ under one of our proposed models. We detail these two terms below.

Discretised distribution Given a predictive distribution obtained by a regression model we can discretise it by using the cumulative density function (cdf). Define $\ell(\hat{q})$ as the real value which we encode quality label $\hat{q}$, as described in Section 3. With this, we obtain the discretised probability mass function

$$p(\hat{q}|\boldsymbol{x}) = \begin{cases} 1 - \text{cdf}(\ell(\hat{q}) - 0.5) & \text{if } \hat{q} = \text{FA} \\ \text{cdf}(\ell(\hat{q}) + 0.5) & \text{if } \hat{q} = \text{Stub} \\ \text{cdf}(\ell(\hat{q}) + 0.5) - \text{cdf}(\ell(\hat{q}) - 0.5) & \text{otherwise,} \end{cases}$$

where the cdf is obtained from the predictive distribution. As we only consider Gaussian distributions for predictions, we can easily use off-the-shelf implementations to obtain the cdf. Figure 4 gives an example of how to obtain the probability of an instance being predicted to be Start (-1.5).

Asymmetric loss function To incorporate asymmetry into the quality label prediction, we define it as

$$\mathcal{L}(\hat{q}, q) = \begin{cases} 0 & \text{if } \hat{q} = q \\ \alpha & \text{if } \hat{q} \neq q, \hat{q} \notin S, q \in S \\ 1 & \text{otherwise} \end{cases},$$

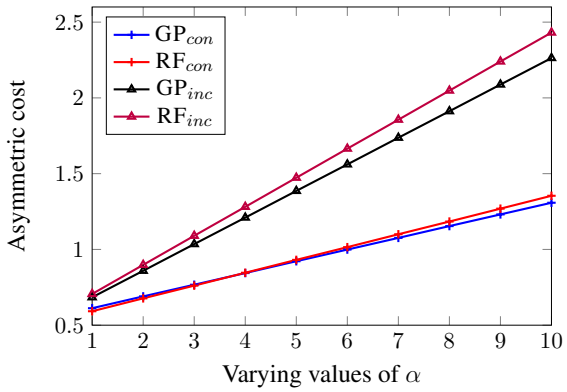

Figure 5: Asymmetric risk vs. α (best viewed in color). Here, GP$_{con}$/RF$_{con}$ denote risk values achieved by GP/RF over *consistent* articles, respectively; GP$_{inc}$/RF$_{inc}$ denote risk values achieved by GP/RF over *inconsistent* articles.

where S is a high-risk label set and $\alpha > 1$ is the penalty associated with a higher risk prediction. In our scenario, we set $S = \{\mathsf{GA}, \mathsf{FA}\}$, as these are the labels we want to give larger penalties when there is a misprediction. Notice that this loss is only an example tailored to our specific setting: other scenarios might warrant different definitions.

Evaluation Under a deployment scenario, one would evaluate $\delta(q)$ for all possible 6 labels and choose the one with minimum risk. However, since in our case we have access to true test set labels, we can just average all $\delta(q)$ for the *gold* labels q in order to assess the models we are interested in (GPs and RFs). As in the previous sections, we report average results over 10 runs for both *consistent* and *inconsistent* test sets.

Results. In Figure 5 we plot the average risk while varying the penalty cost α. As in the intrinsic evaluation result, we see that GPs tend to perform better in the *inconsistent* set. On the other hand, the results in the *consistent* set are very similar, and no conclusions can be made about which method performs best. Overall, the results are favourable towards GPs but the inconclusive results for *consistent* labels shows that there is room for improvements in uncertainty modelling, which we leave for future work.

5 Related Work

The quality assessment of Wikipedia articles is a task that assigns a quality label to a Wikipedia article, reflecting the quality assessment process carried out by the Wikipedia community. Hand-crafted feature-based approaches use features from the article itself (e.g., article length), meta-data features (the number of revisions of an article), and a combination of these two. Various features derived from Wikipedia articles have been used for assessing the quality of Wikipedia articles (Blumenstock, 2008; Lipka and Stein, 2010; Warncke-Wang et al., 2013, 2015; Dang and Ignat, 2016a). For example, Blumenstock (2008) and Lipka and Stein (2010) use article length and writing styles (represented by binarised character trigram features) to differentiate FA articles from non-featured articles, respectively. Warncke-Wang et al. (2015) proposed 11 structural features (such as the number of references and whether there is an infobox or not) to assess the quality of Wikipedia articles. Dang and Ignat (2016a) further proposed nine readability scores (such as the Flesch reading-ease score (Kincaid et al., 1975)) to assess the quality of Wikipedia articles. Based on these last two studies, an online Objective Revision Evaluation Service has been built to measure the quality of Wikipedia articles (Halfaker and Taraborelli, 2015). Features derived from the meta-data of Wikipedia articles — e.g., the number of revisions a Wikipedia article has received — have been proposed to assess the quality of Wikipedia articles (Stvilia et al., 2005; Stein and Hess, 2007; Adler et al., 2008; Dalip et al., 2009, 2017, 2014). For example, Stein and Hess (2007) and Adler et al. (2008) use the authority of editors to measure the quality of Wikipedia articles, as determined by the quality of articles they edited.

Different neural network architectures have been exploited to learn high-level representations of Wikipedia articles. For example, Dang and Ignat (2016b) use a distributed memory version of Paragraph Vector (Le and Mikolov, 2014) to learn Wikipedia article representations, which are used to predict the quality of Wikipedia articles. Dang and Ignat (2017) and Shen et al. (2017) exploit LSTMs (Hochreiter and Schmidhuber, 1997) to learn document-level representations to train a classifier and predict the quality label of an unseen Wikipedia article. Observing that the visual rendering of a Wikipedia article can capture implicit quality indicators (such as images and tables), Shen et al. (2019) use Inception V3 (Szegedy et al., 2016) to capture visual representations, which are used to classify Wikipedia articles based on their quality. They further propose

a joint model, which combines textual representations from bidirectional LSTM with visual representations from Inception V3, to predict the quality of Wikipedia articles.

Beck et al. (2016) explore prediction uncertainty in machine translation quality estimation (QE), where post-editing rate is the dependent variable. In QE, the post-editing rate — which is computed by dividing the post-editing time by the length of the translation hypothesis — is a positive real value. The performance of a GP model was studied in both underestimate and overestimate scenarios. Beck and Cohn (2017); Beck (2017) employ GPs to model text representation noise in emotion analysis, where Pearson's correlation and NLPD are used as the evaluation metrics. Our work is different from these two studies as we model the subjectivity of quality assessment explicitly, which can mimic people's different opinions over the quality of a document.

There is also a rich body of work on identifying trustworthy annotators and predicting the correct underlying labels from multiple annotations (Hovy et al., 2013; Cohn and Specia, 2013; Passonneau and Carpenter, 2014; Graham et al., 2017; Paun et al., 2018). For example, Hovy et al. (2013) propose an item-response model to identify trustworthy annotators and predict the true labels of instances in an unsupervised way. However, our task is to measure the uncertainty of a model over its predictions (as distinct from attempting to learn the "true" label for an instance from potentially biased/noisy annotations), in addition to correctly predicting the gold label, in the context of assessing the quality of Wikipedia articles. Additionally, we have a rich representation of the data point (i.e. document) that we are attempting to label, whereas in work on interpreting multiply-annotated data, there is little or no representation of each data point that has been annotated. Finally, we do not have access to the IDs of annotators across documents, and thus cannot model annotator reliability or bias.

6 Conclusion and Future Work

In this paper, we proposed to measure the uncertainty of Wikipedia article quality assessment systems using Gaussian processes and random forests, utilising the NLPD evaluation metric to measure performance over *consistent* and *inconsistent* articles. Experimental results show that both GPs and RFs are less certain over *inconsistent* articles, where people tend to disagree over their quality, and GPs are more conservative in their predictions over such articles. To imitate a real world scenario where decision-making processes based on model predictions can lead to different costs, we proposed an asymmetric cost, which takes the prediction uncertainty into consideration. Empirical results show that GPs are a better option if overestimates are heavily penalised.

In the future, we are interested in conducting a user study to find out how Wikipedians respond to the utility of uncertainty information provided to them. On the modelling side, having equal intervals between adjacent quality classes is a heuristic, which is potentially inappropriate in the case of Wikipedia. Thus we are also planning to model the quality assessment of Wikipedia articles as an ordinal regression problem, where the gap between adjacent quality labels can vary.

References

B. Thomas Adler, Krishnendu Chatterjee, Luca De Alfaro, Marco Faella, Ian Pye, and Vishwanath Raman. 2008. Assigning trust to Wikipedia content. In *Proceedings of the 4th International Symposium on Wikis*, pages 26:1–26:12.

Daniel Beck. 2017. Modelling representation noise in emotion analysis using Gaussian processes. In *IJCNLP*, volume 2, pages 140–145.

Daniel Beck and Trevor Cohn. 2017. Learning kernels over strings using Gaussian processes. In *IJCNLP*, pages 67–73.

Daniel Beck, Lucia Specia, and Trevor Cohn. 2016. Exploring prediction uncertainty in machine translation quality estimation. In *CoNLL*, pages 208–218.

Joshua E. Blumenstock. 2008. Size matters: Word count as a measure of quality on Wikipedia. In *WWW*, pages 1095–1096.

Leo Breiman. 2001. Random forests. *Mach. Learn.*, 45(1):5–32.

Joaquin Quiñonero Candela, Carl Edward Rasmussen, Fabian H. Sinz, Olivier Bousquet, and Bernhard Schölkopf. 2005. Evaluating predictive uncertainty challenge. In *Machine Learning Challenges*, volume 3944, pages 1–27.

Jeanne Sternlicht Chall and Edgar Dale. 1995. *Readability Revisited: The New Dale-Chall Readability Formula*. Brookline Books.

Krzysztof Chalupka, Christopher K. I. Williams, and Iain Murray. 2013. A framework for evaluating

approximation methods for Gaussian process regression. *Journal of Machine Learning Research*, 14(1):333–350.

Youngmin Cho and Lawrence K. Saul. 2009. Kernel methods for deep learning. In *NIPS*, pages 342–350.

Peter F. Christoffersen and Francis X. Diebold. 1997. Optimal Prediction Under Asymmetric Loss. *Econometric Theory*, 13(06):808–817.

Trevor Cohn and Lucia Specia. 2013. Modelling annotator bias with multi-task gaussian processes: An application to machine translation quality estimation. In *ACL*, pages 32–42.

Edgar Dale and Jeanne S. Chall. 1948. A formula for predicting readability: Instructions. *Educational Research Bulletin*, 27(2):37–54.

Daniel H. Dalip, Marcos A. Gonçalves, Marco Cristo, and Pável Calado. 2009. Automatic quality assessment of content created collaboratively by web communities: A case study of Wikipedia. In *JCDL*, pages 295–304.

Daniel Hasan Dalip, Marcos André Gonçalves, Marco Cristo, and Pável Calado. 2017. A general multiview framework for assessing the quality of collaboratively created content on web 2.0. *Journal of the Association for Information Science and Technology*, 68(2):286–308.

Daniel Hasan Dalip, Harlley Lima, Marcos André Gonçalves, Marco Cristo, and Pável Calado. 2014. Quality assessment of collaborative content with minimal information. In *JCDL*, pages 201–210.

Quang-Vinh Dang and Claudia-Lavinia Ignat. 2016a. Measuring quality of collaboratively edited documents: The case of Wikipedia. In *The 2nd IEEE International Conference on Collaboration and Internet Computing*, pages 266–275.

Quang-Vinh Dang and Claudia-Lavinia Ignat. 2016b. Quality assessment of Wikipedia articles without feature engineering. In *JCDL*, pages 27–30.

Quang-Vinh Dang and Claudia-Lavinia Ignat. 2017. An end-to-end learning solution for assessing the quality of Wikipedia articles. In *The 13th International Symposium on Open Collaboration*, pages 4:1–4:10.

Yarin Gal, Mark van der Wilk, and Carl E. Rasmussen. 2014. Distributed variational inference in sparse Gaussian process regression and latent variable models. In *NIPS*, pages 3257–3265.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2017. Can machine translation systems be evaluated by the crowd alone? *Natural Language Engineering*, 23(1):3–30.

Robert Gunning. 1969. The fog index after twenty years. *International Journal of Business Communication*, 6(2):3–13.

Aaron Halfaker and Dario Taraborelli. 2015. Artificial intelligence service "ore" gives Wikipedians x-ray specs to see through bad edits. *[online] Available: https://blog.wikimedia.org/2015/11/30/artificial-intelligence-x-ray-specs*.

James Hensman, Nicoló Fusi, and Neil D. Lawrence. 2013. Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*.

José Miguel Hernández-Lobato and Ryan P. Adams. 2015. Probabilistic backpropagation for scalable learning of Bayesian neural networks. In *ICML*, pages 1861–1869.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

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

Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. 2011. Sequential model-based optimization for general algorithm configuration. In *5th International Conference on Learning and Intelligent Optimization*, pages 507–523.

J. Peter Kincaid, Robert P. Fishburne Jr., Richard L. Rogers, and Brad S. Chissom. 1975. Derivation of new readability formulas (automated readability index, Fog count and Flesch reading ease formula) for navy enlisted personnel. Technical report, Institute for Simulation and Training, University of Central Florida.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *ICML*, pages 1188–1196.

Nedim Lipka and Benno Stein. 2010. Identifying featured articles in Wikipedia: Writing style matters. In *WWW*, pages 1147–1148.

Alexander G. de G. Matthews, Mark van der Wilk, Tom Nickson, Keisuke. Fujii, Alexis Boukouvalas, Pablo León-Villagrá, Zoubin Ghahramani, and James Hensman. 2017. GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, 18(40):1–6.

Rebecca J. Passonneau and Bob Carpenter. 2014. The benefits of a model of annotation. *TACL*, 2:311–326.

Silviu Paun, Bob Carpenter, Jon Chamberlain, Dirk Hovy, Udo Kruschwitz, and Massimo Poesio. 2018. Comparing bayesian models of annotation. *TACL*, 6:571–585.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher,

Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543.

Carl Edward Rasmussen and Christopher KI Williams. 2006. *Gaussian Process for Machine Learning*. MIT press.

Aili Shen, Jianzhong Qi, and Timothy Baldwin. 2017. A hybrid model for quality assessment of Wikipedia articles. In *Australasian Language Technology Association Workshop*, pages 43–52.

Aili Shen, Bahar Salehi, Timothy Baldwin, and Jianzhong Qi. 2019. A joint model for multimodal document quality assessment. In *JCDL*.

Klaus Stein and Claudia Hess. 2007. Does it matter who contributes: A study on featured articles in the German Wikipedia. In *Hypertext*, pages 171–174.

Besiki Stvilia, Michael B. Twidale, Linda C. Smith, and Les Gasser. 2005. Assessing information quality of a community-based encyclopedia. In *The 2005 International Conference on Information Quality*, pages 442–454.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the Inception architecture for computer vision. In *CVPR*, pages 2818–2826.

Michalis K. Titsias. 2009. Variational learning of inducing variables in sparse Gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pages 567–574.

Hal Varian. 1975. A Bayesian Approach to Real Estate Assessment. *Studies in Bayesian Econometrics and Statistics in Honor of Leonard J. Savage*, pages 195–208.

Morten Warncke-Wang, Vladislav R. Ayukaev, Brent Hecht, and Loren Terveen. 2015. The success and failure of quality improvement projects in peer production communities. In *The 18th ACM Conference on Computer-Supported Cooperative Work and Social Computing*, pages 743–756.

Morten Warncke-Wang, Dan Cosley, and John Riedl. 2013. Tell me more: An actionable quality model for Wikipedia. In *The 9th International Symposium on Open Collaboration*, pages 8:1–8:10.