Lenti: An Adaptive Statistical Approach for Identifying Task-Specific Data Quality Measures

Jeremy Debattista[1]

[1]*No Affiliation*

Abstract

One common aspect between machine learning and interlinking is that both of these data-centric tasks require good quality datasets. Finding the right quality criteria that best describe the data required for the task at hand is particularly challenging. Furthermore, different data consumers will have different quality criteria for the same task at hand. In this article, we present a novel approach to assist data consumers to identify key quality indicators for particular tasks. Based on user feedback, task-specific models learn and adapt these indicators and weights. We apply this algorithm in a dataset retrieval portal for a digital library use case, used to find external datasets for creating metadata in a digital library to evaluate the relevance and performance of our proposed approach. Overall we show that our approach gives a precision of 0.81 when suggesting key quality indicators for a specific task.

Keywords

Data Quality, Statistical Models, Quality Measures

1. Introduction

The digital transformation of businesses and organisations in a data-driven world is reflected in the renewed importance of *data quality*. The growth in the number of datasets freely available and the abundance of computational resources has made it easier for this digital transformation to happen, and deliver new and innovative services to the end-user.

Data is now more easily accessible, with portals such as Google dataset search[1] and Kaggle[2] making large datasets available for use with a click of a button. Whilst all of this is positive, one cannot deny that the quality aspects of these datasets are often ignored by data consumers due to the lack of quality information or metadata available for each individual item. Data consumers are using these datasets "blindly", without considering the potential implications if the data used is of poor quality. Poor data quality leads to technical and social implications.

A recent citable example [1] is the case of COMPAS, a risk assessment software used by the US courts that forecasts which criminals are most likely to re-offend in the future. This software was being used to help judges in their sentences. In their analysis [1], Angwin et al. flagged that the software was racially biased, where African-American offenders were seen to be almost twice more likely to re-offend in the future, and hence labeled as being of a higher

MEPDaW'23: Managing the Evolution and Preservation of the Data Web, November 06, 2023, Athens, Greece
jerdebattista@gmail.com (J. Debattista)
0000-0002-5592-8936 (J. Debattista)

[1]https://toolbox.google.com/datasetsearch
[2]https://www.kaggle.com/

risk to society than their white counterparts. This example demonstrates that in AI applications it is important that training datasets are of high quality, as poor quality data will propagate errors throughout the system, and ultimately reach the end-user.

Data quality is not only important for AI, but it can be applied to any other data-centric domain. In Linked Data, for example, it is important that task-appropriate quality datasets are used for interlinking with external knowledge graphs. For example, if we are building a Linked Data-based question answering system, it would be pointless to create interlinks with an online knowledge graph that has insufficient availability, even if it is the most trustworthy or deemed to be the most complete for additional explanation or inference.

The motivation behind our work is therefore to lower the barriers to using data assets by assisting data consumers in identifying the important quality measures for the task at hand. With existing quality assessment tools providing metadata on a data asset, an algorithm that can help identify the right, or most important quality measures, will create new-found opportunities in the area of dataset retrieval, data lakes, recommender systems, and data governance amongst other applications. To this end, we define the following research question:

> *To what extent can data consumers be supported in identifying the right quality measures for a specific task?*

In this article we present a novel approach based on statistical models that learns from what previous users (both experts and non-experts) chose as quality measures for a particular task. Having this knowledge, the model is then able to suggest what quality measures are important for the specific task with a degree of confidence. Furthermore, the algorithm also suggests relative importance weights for the suggested quality measures.

The main contributions of this article are:

1. A formal definition of a user-based feedback learning approach for suggesting quality measures and importance weights for specific tasks (Section 3);
2. An evaluation of the model and its application in a dataset retrieval portal within library science domain (Section 4).

Section 2 gives an overview of the current state of the art approaches, whilst the conclusions are described in Section 5.

2. Related Work

To the best of our knowledge, there is no comparable work that assists data consumers identify the right quality measures for a particular task at hand. In this section we therefore discuss relevant work that considers information quality as a central aspect in their systems.

Tejda-Lorente et al. [2] proposed a recommender system that takes into account the item's quality as a new factor in the recommendation process. The recommender system was evaluated in a digital library scenario, where the aim was to ease the information overload provided by digital libraries, filtering and discovering the right information and presenting it to students and staff. Tejda-Lorente et al. show that when quality was taken into consideration the mean average error was lowered by 4.8%, meaning that the quality-driven recommendations were closer to the users' preferences. Whilst these are promising results, this approach does not consider different tasks that a user might be performing, hence potentially requiring different quality aspects for the recommended items.

Literature on quality-driven filtering and ranking of datasets is more common than quality-driven recommender systems. In [3], users can rank datasets by selecting a number of quality measures at different quality granularity (category, dimension or metric) and then assigning weights. This selection is then used within the user-weighted ranking algorithm to rank datasets using their quality metadata. A similar ranking process can be observed in Färber et al.'s knowledge graph recommendation framework [4]. The WIQA framework [5] was designed to allow users to create and apply policies based on indicators such as provenance and background context related to data providers to filter information in named graphs. Bizer and Cyganiak go a step further with WIQA, where they provide explanation on the resulting set of filtered triples.

The identification of quality measures is not only relevant for data retrieval or recommendation. Where quality metadata is not available, data consumers might need to assess the quality of certain datasets themselves. Whilst identification of quality measures is also a major step in any data quality methodology [3, 6], data quality assessment is an expensive process. Therefore, the proposed approach can help users in identifying the right measures required for quality assessment for a particular task.

3. Lenti - A user-based feedback learning approach

In this section we provide a formal definition for Lenti, an approach for helping users identify the right quality measures for a given task. However, prior to discussing our approach, we provide some background definitions as used throughout the rest of the article.

3.1. Background Definitions

Quality Profile - Describes quality measures and weights together with an identification of a task. In terms of descriptive statistics, a *quality profile* is an individual with an observation for each quality measure in the profile;

Key Quality Indicators (KQI) - Sometimes also referred to as Key Quality Measures, is a set of quality measures that are identified as important for a given task;

Importance Weight - A value between 0 and 100 inclusive (or 0.0 and 1.0 inclusive) assigned to KQIs. This could be used in ranking functions to favour higher weighted measures than others;

Proportion Score - A default, equal score given to all quality measures for a particular task used as the hypothesis test condition;

Popularity Score - A task-specific individual score given to all the quality measures based on all the observations from the quality profiles, which is used for the hypothesis testing;

Filtering Threshold - *P-value* in statistical terms, this threshold is used to identify whether a quality measure is statistically significant, hence a key quality indicator, for a particular task.

3.2. Formal Definition of Lenti

The method that we propose takes inspiration from a well known technique in machine learning and statistics; *feature selection*. Whilst there are a number of ways of implementing this technique, most algorithms typically use a target variable (or class) to test against. In our approach, we need to identify a filtering threshold for the quality measures in a *quality profile*. Therefore, in this regard, our idea is to define the *proportion score* for each quality measure identified in a profile. Using the quality profile data that is fed to the algorithm, for each quality measure we calculate the *popularity score* and check how much it differs from the given *proportion score*. This technique is used to reduce the quality measures to the most important one for a particular task. The given measures are subsequently used to identify an *importance weight*, which together are then presented to the user who can either modify and re-train the models within Lenti or use them to filter and rank datasets in a quality driven data portal. Lenti creates a statistical model to represent the different tasks, as different tasks would have different key quality indicators and importance weights.

3.3. The Quality Profile Vector

We represent a quality profile individual as a row vector:

$$\vec{q_x} = \begin{bmatrix} q_1 & q_2 & \cdots & q_{i-1} & q_i \end{bmatrix} \tag{1}$$

where $\vec{q_x}$ refers to the vector for the quality profile x, and q_i, $1 \leq i \leq n$ is the observation value (or the assigned weight) between 0 and 1 (both numbers inclusive) of a given quality measure, and n is the number of quality measures in a profile.

To build a quality profile vector, each quality measure is assigned a position in the vector using a defined function $pos : f(x) \rightarrow \mathbb{P}$, where $f(x)$ is a function, for example a hash function or ordering function, that given a particular quality measure string identifier (e.g. Accessibility), it will return a non-negative integer $\mathbb{P} \in [0, n)$ that is mapped to a column in the quality profile. For simplicity, we define the inverse function $pos^{-} : g(\mathbb{P}) \rightarrow x$, where given the non-negative integer $\mathbb{P}$, the function g returns the quality measure string identifier. A specific task has more than one quality profile, therefore we represent all quality profiles of *one* task in an $m \times n$ matrix:

$$\mathbf{T} = \begin{bmatrix} \vec{q}_{11} & \vec{q}_{12} & \cdots & \vec{q}_{1n-1} & \vec{q}_{1n} \\ \vec{q}_{21} & \vec{q}_{22} & \cdots & \vec{q}_{2n-1} & \vec{q}_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \vec{q}_{m1} & \vec{q}_{m2} & \cdots & \vec{q}_{mn-1} & \vec{q}_{mn} \end{bmatrix} \tag{2}$$

where, m is the number of observations (i.e. the total number of profiles) and n is the total number of quality measures in the matrix. If a value does not exist for any position $\mathbf{T}_{i,j}$, then that position in the matrix is filled with a 0.

3.4. Filtering Quality Measures

In machine learning, identifying the relevant features of a dataset is an important step to prevent overfitting and ensure generalisation. This is usually done through supervised feature selection techniques, whereby the data predictor variables are tested against a target class in order to, for example, find which features maximise relevancy and minimise redundancy [7]. We apply the idea of feature selection to our problem to find the most relevant quality measures for a particular task, however, our input matrix $\mathbf{T}$ has no target class. In this case we need to identify a suitable filtering threshold in order to remove "features", in our case quality measures, from $\mathbf{T}$ that are deemed as not being important for the task. We favoured a feature selection approach over a dimensionality reduction approach as the former keeps a subset of the original quality measures, whilst the latter will create new components made up from the original measures.

To solve the problem of not having a target class, we investigate the use of *z-score for population proportions*, in order to determine the popularity of the quality measures among each other, and hence identifying the key quality measures. The first step towards building a model for a specific task in Lenti is to assign each quality measure for the specific task an equal *proportion score* p as:

$$p = \frac{1}{n} \tag{3}$$

where n is the number of quality measures (or columns) in a matrix $\mathbf{T}$. This score will act as our baseline to compare the popularity between the quality measures, and thus we identify the following hypotheses to test:

H_0**:** Quality Measure qm_x is not more popular than the average proportion of $p \leq \frac{1}{n}$

H_1**:** Quality Measure qm_x is more popular than the average proportion of $p > \frac{1}{n}$

Hypothesis testing is performed in order to identify whether to accept the null hypothesis (H_0) and thus consider the quality measure qm_x as not being an important measure for the particular task, or otherwise.

Having the hypotheses to test, the z-score is used to calculate the *popularity score* for each qm_x over a matrix $\mathbf{T}$. Therefore, we define a function *popularity* $: h(qm_x) \rightarrow \mathbb{R}$ as follows:

$$\begin{aligned} popularity(qm_x) &= \frac{(\widehat{p}_x - p)}{\sqrt{\frac{p \times (1-p)}{m}}} \\ \widehat{p}_x &= \frac{qm(x)}{\mathbf{T}_{\sum \mu}} \\ \mathbf{T}_{\sum \mu} &= \sum_{j=0}^{n-1} qm(pos^{-}(j)) \\ qm(x) &= \frac{\sum_{i=0}^{m-1} \mathbf{T}_{i,pos(x)}}{m} \end{aligned} \tag{4}$$

where x is the string identifier of the quality measure (e.g. Accessibility), m is the number of observations (or rows) in the matrix $\mathbf{T}$. $\mathbf{T}_{\sum \mu}$ is the total of all means of the given observations in a matrix $\mathbf{T}$, which always add up to 100 (or 1.0 if the weights are between 0 and 1 inclusive, instead of 0 and 100 inclusive), whilst $\widehat{p_x}$ returns the proportion score of a quality measure x based on the given observations in matrix $\mathbf{T}$.

Once the *popularity score* is calculated, the score is normalised. This enables us to test whether we should reject or accept the null hypothesis, hence identify which quality measures should be filtered out. Therefore, given a task and its corresponding matrix $\mathbf{T}$, the popularity scores, and a filtering threshold value ϑ:

$$\begin{aligned} M_{\mathbf{T}} &= \{qm_x \mid \forall qm_x \in QM_{\mathbf{T}} \cdot popularity(qm_x) < \vartheta\} \\ QM_{\mathbf{T}} &= \{pos^{-}(i) \mid \forall i \cdot 0 \leq i < n\} \end{aligned} \tag{5}$$

where $QM_{\mathbf{T}}$ is the set of all quality measures for the given task. The threshold ϑ follows the asymptotic significance (also known as the p-value[3]) in statistical hypothesis testing, where a lower value suggest stronger evidence to reject or accept the null hypothesis, and $M_{\mathbf{T}}$ represent the set of important quality measures for the given task. By default, we set the ϑ value to 0.01.

3.5. Defining the Importance Weight

Identifying the relevant quality measures for a particular task is not enough to identify which datasets are fit for one's needs. Some measures might have a higher importance than others and thus have to be reflected in the algorithm's suggestion. The most straightforward way would be to take the mean value of all observations for a particular quality metric, however, given that the observations might not follow a normal distribution, the resulting *importance weight* can be overcompensating towards outliers. The median value gives a more representative value of the observations central tendency within a skewed distribution, moving towards the mean value if future observations grow into a normal distribution. Keeping in mind that future observations will be propagated to task models, we apply an incremental median estimator, in a similar fashion as defined by Feldman and Shavitt in [8] in order to identify the importance weight. Feldman and Shavitt describe and formally verify an algorithm, FAME, that estimates the median value of internal Internet link delays. The main idea of FAME is to decrease the required storage space to two variables, the current median estimator value, and a step variable that indicates how far a value should move to the next median estimate. The authors claim that the algorithm converges to an accurate median with increasing data.

Therefore, starting from the first quality profile in the matrix $\mathbf{T}$, we iteratively estimate the median over all profiles, and thus based on [8] we define the incremental median approximation as follows:

$$M_{qm(x)} = M'_{qm(x)} + \varepsilon_{qm(x)} \times sgn(\vec{q_{p+1}}_{pos(x)} - M'_{qm(x)}) \tag{6}$$

where $M'_{qm(x)}$ is the current estimated median (0 if it is the first observation), $\varepsilon_{qm(x)}$ is the approximation parameter (defined as step variable in [8]) for a specific quality measure, $\vec{q_{p+1}}_{pos(x)}$ is the weight for quality measure x in the current quality profile being observed, and sgn is the signum function.

[3]Note: not to be confused with the defined *proportion score p*.

The incremental median estimation approach defined in [8] was considered at this stage as opposed to the traditional median function so that the estimation will have enough data to converge the $\varepsilon_{qm(x)}$ value to more accurate estimations. The $\varepsilon_{qm(x)}$ is initially set to the maximum value of an arbitrary value ($\mathbb{R} \in (0,\ 1]$) and the half of the first weight encountered for that measure. The initial arbitrary value is required, otherwise if the first weight encountered is 0, then $M_{qm(x)}$ will always be 0.

3.6. Propagating User's Feedback

Following the initial training of the algorithm, the system will continue to train itself whenever a user provides feedback to the suggestions provided by Lenti. The propagation of user feedback encapsulates the true meaning of *fitness for use*, as Lenti will update itself with new knowledge about the potential quality measures for a specific task, and hence continue shaping the right measures for the said task.

Propagating new user feedback will affect both the quality measure filtering and the importance weight. For this we need to use the *incremental mean* estimator to propagate the *popularity score* and the *incremental median* estimator for the new *importance weight*. User feedback is treated as a quality profile and is represented in a row vector ($\vec{q_{p+1}}$) as described in Equation 1. This is added to matrix **T**, incrementing *m* (i.e. the total number of rows in **T**). The function *popularity*(qm_x) defined in Equation 4 is recalculated using the incremented values for $_{qm(x)}$ (Equation 7). The incremental mean is calculated as follows:

$$_{qm(x)} = {}'_{qm(x)} + \frac{\vec{q_{p+1}}_{pos(x)} - {}'_{qm(x)}}{m} \tag{7}$$

where ${}'_{qm(x)}$ is the previously calculated mean value for quality measure x, $\vec{q_{p+1}}_{pos(x)}$ is the new weight for quality measure x assigned during the user feedback, and m is the total number of observation values (i.e. total number of quality profiles for that task). This leads to the identification of potentially new key quality measures for the task and new importance weights for the identified measures using the incremental median estimator (Equation 6).

4. Evaluation

In order to evaluate Lenti, we implement a quality-based dataset retrieval portal for librarians working in a digital library[4]. Using this portal, we perform two experiments. For the first experiment we look at the relevancy of the suggested key quality indicators for a particular task. For the second experiment, we evaluate the ranked datasets given by the portal, comparing the results returned following a ranking configured with Lenti suggested quality measures for the given task, against the evaluators choice of quality measures for the same task.

4.1. Library Science Use Case

Information professionals (IPs) are always striving to improve the quality of their digital libraries, ensuring the availability of high quality metadata. The success of a digital library is usually

[4]Proof-of-concept available here: https://github.com/jerdeb/qualityrecommender

measured via the quality of its metadata [9]. This means that information professionals have to make sure that they follow standards and that the external metadata used from controlled vocabularies (such as the Library of Congress) is also of high quality. Following a discussion with 3 librarians at the authors' host university's library, they agreed that most of these choices are taken following years of experience and trusting certain authorities. However, these choices sometimes are also based on guidelines set by the institution.

4.1.1. Evaluation Task.

One aspect that is gaining momentum within IPs and digital libraries is the use of Linked Data[5]. IPs realised that Linked Data offers many benefits, such as better resource discovery and interoperability [10]. Therefore, in order to support IPs in the task of creating external interlinks in bibliographic metadata, a quality-based dataset retrieval portal was implemented with a number of linked datasets pre-assessed and their metadata added to the portal. Lenti is used to suggest quality measures and importance weights to IPs who do not yet have a clear definition of the quality aspects required before choosing an external dataset.

Each participant of the evaluation was given the following task:

> **You are creating bibliographic metadata in a Linked Data format for items in your digital library. As per Linked Data principles, you require to link your metadata with other bibliographic datasets and/or controlled vocabularies.**

For Lenti, this translates to a *bibliographic metadata interlinking* task.

4.1.2. Data Collection.

In order to conduct this evaluation and build the quality-based driven portal we need to collect both primary and secondary data.

With regard to *primary data*, we conducted a questionnaire among information professionals asking what quality measures they consider to be important when using or searching for external data sources to interlink to when creating bibliographic metadata. The answers were extracted and transformed into quality profiles and used within the Lenti as training data. In total we had 158 participants, from which 142 quality profiles were extracted.

Secondary data included the identification of bibliographic and controlled vocabulary linked datasets, domain specific quality metrics, and quality metadata from the assessed datasets. For the identification of datasets we downloaded available datasets from the LOD cloud[11] that were tagged with "Publications" and other specific datasets mentioned in [10]. These led us to the identification of 25 linked datasets, 16 of which are controlled vocabulary linked datasets, 8 are bibliographic datasets, and DBpedia. With regard to quality metrics, we followed upon the survey described by Debattista et al. [12] extracting library science specific quality metrics. In total, 18 metrics were implemented from 11 different dimensions, to which we refer to as the

[5]https://www.oclc.org/research/themes/data-science/linkeddata/linked-data-survey.html Last Access Date: 9th April 2019

	Suggested	Not Suggested
Relevant	27	36
Irrelevant	6	52

Table 1
Confusion Matrix capturing the relevancy of suggested metrics

quality measures for simplicity. The identified 25 datasets were assessed over these metrics and quality metadata was produced. The quality metadata was used to rank datasets.

4.2. Evaluating Lenti

Precision and **recall** are two widely used metrics to evaluate statistical models. The **precision** *Pr* (Eq. 8) measures the probability that a recommended quality measure is relevant to the user, therefore the metric calculates the ratio between the number of relevant suggestions to the number of suggested items.

$$Pr = \frac{\#Suggested\ and\ Relevant\ QM}{\#Number\ of\ Suggested\ QM} \tag{8}$$

On the other hand, the **recall** *R* (Eq. 9) metric measures the probability that a relevant quality measure is recommended to the user, therefore, the metric calculates the ratio between the number of relevant suggestions to the number of all relevant quality measures.

$$R = \frac{\#Suggested\ and\ Relevant\ QM}{\#Number\ of\ Relevant\ QM} \tag{9}$$

4.2.1. Experiment 1: Relevancy of Suggested Quality Measures

In this experiment we asked the participants to choose the most appropriate quality measures for the Evaluation Task that was previously outlined. The measures that were suggested by Lenti were highlighted to the participants, who were also provided with a list of the rest of the quality measures.

Table 1 represents a confusion matrix that shows the frequency of suggested and not suggested measures, as well as if the measures were considered to be relevant or irrelevant by the participants. Lenti has been evaluated by 11 participants. This resulted into a precision (*Pr*) value of 0.81 and a recall value (*R*) of 0.43.

Our approach demonstrates a high precision value. This was expected as the threshold value by default is set very low (0.01), therefore a quality measure is considered to be a key quality indicator for a specific task only if there is strong evidence to reject the null hypothesis and to accept the alternative hypothesis (H_1 cf. Section 3.4). Furthermore, given the restrictive selection nature of the algorithm and the participants' data quality needs, it was expected that the system will demonstrate low recall values. As expected, all participants, except for participant number 6, indicated other quality measures to be relevant along with those suggested by Lenti. In Table 2 we breakdown the results for each participant. Six participants chose all three suggested measures, whilst four chose at lease two suggested measures.

Participant	1	2	3	4	5	6	7	8	9	10	11
Precision	1.00	0.67	0.67	1.00	0.33	1.00	1.00	0.67	1.00	0.67	1.00
Recall	0.50	0.33	0.40	0.50	0.25	1.00	0.43	0.33	0.38	0.33	0.50

Table 2
Precision and recall per participant

4.2.2. Experiment 2: Relevancy of Ranked Datasets

The aim of this evaluation is to compare the ranked datasets given by the portal using the quality measures and importance weights (configuration) *suggested by Lenti*, against the same datasets ranked using the configuration *chosen by the evaluators*. In both cases, the ranking of the datasets depends on the quality assessment values defined in the datasets' quality metadata. Therefore, the same dataset with a specific quality assessment value can be ranked differently for both the above configurations. The participants did not know the quality of the assessed datasets, therefore they had no knowledge whether data publishers favoured particular quality measures than other when producing and publishing the dataset.

Once the configuration for the quality measures and importance weights are set, the portal ranks the datasets, showing the top 6 ranked datasets to the participants. Participants then had to identify the relevant datasets. For the scope of this experiment we use the mean average precision at cutoff N (MAP@N) metric. We use this metric as we consider the order of the ranked items to be important.

In order to calculate the mean average precision, we need to calculate the average precision (AP) first. The average precision is measured by taking the precision scores for each relevant retrieved item [13]. In our case, we are only interested in the 6 top ranked datasets, therefore, unlike the AP measure, the AP@N formula (Eq. 10) takes into consideration only a set number of items instead of all potential relevant items. AP@N is measured as follows:

$$AP@N = \frac{1}{m} \sum_{k=1}^{N} P(k) \times rel(k) \tag{10}$$

where m is the total number of relevant items in the retrieved spaced, N is the total number of items that have to be retrieved (i.e. the cutoff), $P(k)$ is the precision value at cut-off k, and $rel(k)$ gives 1 if the item at rank k is relevant or 0 otherwise.

The mean average precision (MAP) [13] is usually used in information retrieval system, where ranking is important, to average its precision over a number of queries. In our case, rather than number of queries, we will average the precision over the number of participants (u). The mean average precision is defined as follows [13]:

$$MAP = \frac{1}{u} \sum_{u} AP@N_u \tag{11}$$

The evaluation shows a low MAP@N score for both ranking sets using the two configurations. For the Lenti configurations we report a value of 0.376, whilst for the participants' configurations a value of 0.386. In terms of comparison between the two rankings, whilst the participants' configuration gave a higher MAP@N value, the difference can be considered to be insignificant.

This low score can be contributed to either (1) the fact that the evaluators were expecting to chose data sources that they are usually comfortable working with, rather than those data sources which were considered of better quality; or (2) the datasets that the evaluators expected to rank high had poor quality attributes for the chosen configuration (quality measures and importance weights), hence ranking lower. For example, only two out of the 10 most frequently used datasets for interlinking mentioned in McKenna et al. [10] are ranked in the top 6 datasets using Lenti's suggested measures, namely Geonames (top ranked) and the Library of Congress datasets (third ranked).

5. Conclusions

In this article we proposed and presented Lenti as our key contribution. Lenti's approach is to create task-specific statistical models that learn and update from past user interaction in order to identify and suggest the key quality measures and their importance weights for a specific task. This approach can be used to support or within various data consumer centric applications, such as recommender systems and dataset retrieval portals.

Our evaluation confirms that Lenti favours precision in the identification of quality measures relevant for a specific task (precision value of 0.81), as opposed to a potentially suggesting quality measures that are not relevant for the task at hand (recall value of 0.43).

In the second part of the evaluation the Lenti suggestions were used within the dataset retrieval main function, that is, finding and ranking datasets. Here, the MAP@N results did not give us satisfactory results. Whilst the choice of measures and importance weights affected the ranking score, the MAP@N score takes into consideration the order of the ranking of the datasets that is also dependent on their quality. Nonetheless, we show that there was insignificant difference between the Lenti suggested measures and importance weights, and the participants' choice.

Overall we show that given a set of quality profiles for a task, our proposed approach can assist data consumers in identifying the right quality measures. Nonetheless, we have to highlight the limitations of our findings; mainly that we only had a total of 11 participants evaluating the approach and it was limited to just one task. The end goal of Lenti is to learn *on-the-fly* without having to know about a potentially new task. The premise of our approach is that statistical models will learn the KQIs for different tasks based on usage experience

References

[1] J. Angwin, J. Larson, S. Mattu, L. Kirchner, Machine Bias. There's software used across the country to predict future criminals. And it's biased against blacks, ProPublica.org (2016). URL: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

[2] Á. Tejeda-Lorente, C. Porcel, E. Peis, R. Sanz, E. Herrera-Viedma, A quality based recommender system to disseminate information in a university digital library, Inf. Sci. 261 (2014) 52–69. URL: https://doi.org/10.1016/j.ins.2013.10.036. doi:10.1016/j.ins.2013.10.036.

[3] J. Debattista, S. Auer, C. Lange, Luzzu – a methodology and framework for Linked Data quality assessment, Data and Information Quality 8 (2016).

[4] M. Färber, F. Bartscherer, C. Menne, A. Rettinger, Linked data quality of dbpedia, freebase, opencyc, wikidata, and YAGO, Semantic Web 9 (2018) 77–129. URL: https://doi.org/10.3233/SW-170275. doi:10.3233/SW-170275.

[5] C. Bizer, R. Cyganiak, Quality-driven information filtering using the wiqa policy framework, Web Semant. 7 (2009) 1–10. doi:10.1016/j.websem.2008.02.005.

[6] A. Rula, A. Zaveri, Methodology for assessment of Linked Data quality, in: M. Knuth, D. Kontokostas, H. Sack (Eds.), Proceedings of the 1st Workshop on Linked Data Quality co-located with 10th International Conference on Semantic Systems, LDQ@SEMANTiCS 2014, Leipzig, Germany, September 2nd, 2014., volume 1215 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2014. URL: http://dblp.uni-trier.de/db/conf/i-semantics/ldq2014.html.

[7] H. Peng, F. Long, C. H. Q. Ding, Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy, IEEE Trans. Pattern Anal. Mach. Intell. 27 (2005) 1226–1238. URL: https://doi.org/10.1109/TPAMI.2005.159. doi:10.1109/TPAMI.2005.159.

[8] D. Feldman, Y. Shavitt, An optimal median calculation algorithm for estimating internet link delays from active measurements, in: K. Saraç, T. Friedman (Eds.), Fifth IEEE/IFIP Workshop on End-to-End Monitoring Techniques and Services, E2EMON 2007, 21st May, 2007, Munich, Germany, IEEE Computer Society, 2007, pp. 1–7. URL: https://doi.org/10.1109/E2EMON.2007.375318. doi:10.1109/E2EMON.2007.375318.

[9] A. Tani, L. Candela, D. Castelli, Dealing with metadata quality: The legacy of digital library efforts, Information Processing & Management 49 (2013) 1194 – 1205. URL: http://www.sciencedirect.com/science/article/pii/S0306457313000526. doi:https://doi.org/10.1016/j.ipm.2013.05.003.

[10] L. McKenna, C. Debruyne, D. O'Sullivan, Understanding the position of information professionals with regards to linked data: A survey of libraries, archives and museums, in: Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries (JCDL 2018), Fort Worth, Texas, USA, June 3rd-7th, 2018., 2018, pp. 7–16.

[11] J. P. McCrae, A. Abele, P. Buitelaar, R. Cyganiak, A. Jentzsch, V. Andryushechkin, Linked open data cloud, 2019. URL: http://lod-cloud.net.

[12] J. Debattista, L. McKenna, R. Brennan, Understanding information professionals: A survey on the quality of linked data sources for digital libraries, in: H. Panetto, C. Debruyne, H. A. Proper, C. A. Ardagna, D. Roman, R. Meersman (Eds.), On the Move to Meaningful Internet Systems. OTM 2018 Conferences - Confederated International Conferences: CoopIS, C&TC, and ODBASE 2018, Valletta, Malta, October 22-26, 2018, Proceedings, Part II, volume 11230 of *Lecture Notes in Computer Science*, Springer, 2018, pp. 537–545. URL: https://doi.org/10.1007/978-3-030-02671-4_32. doi:10.1007/978-3-030-02671-4_32.

[13] L. Liu, M. T. Özsu (Eds.), Encyclopedia of Database Systems, Springer US, 2009.