Predicting Quality of Crowdsourced Annotations Using Graph Kernels

Archana Nottamkandath, Jasper Oosterman, Davide Ceolin, Gerben De Vries, Wan Fokkink

Predicting Quality of Crowdsourced Annotations using Graph Kernels

Archana Nottamkandath[1], Jasper Oosterman[2], Davide Ceolin[1], Gerben Klaas Dirk de Vries[3], and Wan Fokkink[1]

[1] VU University Amsterdam, Amsterdam, The Netherlands
{a.nottamkandath,d.ceolin,w.j.fokkink}@vu.nl
[2] Delft University of Technology, Delft, The Netherlands
j.e.g.oosterman@tudelft.nl
[3] University of Amsterdam, Amsterdam, The Netherlands
g.k.d.devries@uva.nl

Abstract. Annotations obtained by Cultural Heritage institutions from the crowd need to be automatically assessed for their quality. Machine learning using graph kernels is an effective technique to use structural information in datasets to make predictions. We employ the Weisfeiler-Lehman graph kernel for RDF to make predictions about the quality of crowdsourced annotations in Steve.museum dataset, which is modelled and enriched as RDF. Our results indicate that we could predict quality of crowdsourced annotations with an accuracy of 75%. We also employ the kernel to understand which features from the RDF graph are relevant to make predictions about different categories of quality.

Keywords: Trust, Machine learning, Crowdsourcing, RDF Graph Kernels

1 Introduction

Cultural Heritage institutions are digitizing their collections. This process involves manually making digital copies of the artifacts in their collection and registering relevant information about the metadata of the artifacts into their systems. Professionals are employed by the institutions to provide this information according to their high quality standards.

In most cases, providing such information for large collections is an exhaustive task in terms of human resources and requires expertise knowledge from many domains. Hiring more professionals with domain expertise in order to speed up the tasks is not feasible, so these institutions are looking into crowdsourcing this artwork description (annotation). The crowd provides diversified information about artifacts, hence issues dealing with the quality of annotations arise. Consider, for instance, the artwork collection item (a sculpture) from Steve.museum depicted in Figure 1; the figure includes the annotations produced by crowd annotators in a real-world annotation campaign. The annotations in green indicate the ones which were considered useful by the professionals at institution while

the red ones indicate the ones which were not considered useful to be added to their collection. Employing human reviewers to assess the quality of annotations is as expensive as hiring professional annotators and thus there is a need for automated processes to assess the quality of these annotations or, in other words, to develop methods to estimate the trust in them.

Fig. 1. The artwork titled *Kinarra* from the Steve.museum dataset and associated crowd annotations. Green = useful, red = non-useful.

Properties of the annotations such as annotator, annotated artifact, time stamp etc. and properties of the artifact and of the annotators themselves can all be modeled using the Resource Description Framework (RDF), i.e. as a labeled graph. Apart from representing the entities and the relations between them, such an RDF graph also captures the structural information of the information. In an earlier work [3], we modelled the annotations and employed some annotation properties such as semantic similarity and reputation of the users to predict the quality of annotations. Machine learning techniques such as Support Vector Machines (SVMs) can be used to make predictions about features in the dataset. Recently, machine learning using graph kernels has arisen as an efficient method for learning from RDF graphs [13,6], that can effectively exploit the structural properties of the graph using SVMs. To show the potential of such a graph kernel we apply it on the Steve.Museum dataset. First we transform the annotations and contextual information from the dataset to a semantic model and enrich the model with external vocabularies and knowledge sources. We then leverage this model to make predictions about the annotation quality by applying the Weisfeiler-Lehman RDF graph kernel.

Our contributions are threefold; 1) We propose a workflow to transform and enrich Cultural Heritage datasets into semantic (RDF) data; 2) We show how a specialized kernel for RDF can be applied on a semantic Cultural Heritage annotation dataset to predict annotation quality and relevant features; and 3) We provide insights into the benefit of RDF kernel for Cultural Heritage datasets.

The paper is structured as follows. In Section 2 we describe related work. In Section 3 we describe the overall workflow and explain in detail about the enriched semantic model and RDF kernel. Section 4 describes the Steve.Museum dataset and the metrics used, followed by the results and their analysis in Section 5. We provide discussion and future work in Section 6.

2 Related Work

Utilizing knowledge from the crowds to perform tasks is widely used on the Web [7]. Open Mind Common Sense [23] is a knowledge acquisition system designed to acquire commonsense knowledge from the general public over the Web. Several Cultural Heritage institutions have been looking towards users on the Web to provide information about their artifacts such as depicted visuals, meta data, sentiments etc. These institutions define tasks for gathering annotations from the users either as a game as in ESP game [25] or through online systems as shown in examples from "Your Paintings Tagger" by BBC[4] Accurator for Rijksmuseum Amsterdam[5], and others such as Brooklyn Museum, New York Library and others [17].

We consider the estimation of quality of crowdsourced annotations as a task equivalent to the estimation of the trustworthiness of the annotations, and indirectly of the trustworthiness of the annotator. We refer the reader to the works of Artz and Gil [1] for an extensive survey of trust models in the Semantic Web, Golbeck [9] for trust models on the Web, Sabater and Sierra [18] for trust models in computer science and Prasad et al. [15] for Bayesian computational trust models.

Studies have been done to understand the quality of information provided by the crowd as shown by Snow et al. [24]. Inel et al. [11] have been studying the annotations obtained from the crowdsourcing platforms such as Crowdflower to make quality assessments. There have also been many methods developed to determine the quality of these crowdsourced information, where majority voting has been widely used. For example, in ESP game, a label is added to the picture if at least two randomly picked users suggest the same label. This research extends two previous works of ours. We extend a Semantic Web representation of cultural heritage annotations that we previously introduced [3], and we explore how to make machine learning-based quality assessments from such a model. These machine learning-based assessments implicitly introduce a measure of similarity between Semantic Web data. The use of semantic similarity measures to semi-automatically predict the quality of crowdsourced cultural heritage annotations has been explored in another previous work of ours [2]. However, in that work semantic similarity is computed only between the annotations, while here it extends to the metadata.

In this paper we utilize RDF graph kernels to utilize structural properties of graphs to make predictions about annotation quality. Although features about

[4] http://tagger.thepcf.org.uk/
[5] http://rma-accurator.appspot.com

the user and of the annotations were used to make predictions of quality with SVMs in a previous work of ours [14], we did not employ RDF graph kernels for the predictions. This paper aims to provide a new method employing RDF graph kernels for automatically predicting quality of crowdsourced annotations in the cultural heritage domain.

3 Approach

In this section we describe the workflow that we propose to assess the quality of crowdsourced annotations. We begin with an overview of the workflow and then we describe each component in detail.

3.1 Workflow Overview

The workflow that we adopt to estimate the quality of the user-provided annotations is depicted in Fig. 2 and consists of three steps:

1. Representing Annotations in RDF
2. Annotations Enrichment
3. Machine learning with graph kernels for RDF

Whenever an annotation is introduced in the system, it is modeled in RDF, along with its related metadata (e.g., its author). The resulting RDF graph is then enriched by linking it with information provided by authoritative and trusted Linked data sources. In this manner, we expand the knowledge graph describing the annotation. Lastly, we use Support Vector Machines and the Weisfeiler-Lehman graph kernel to estimate the quality of the annotation, exploiting the information provided in the enriched graph and using a set of previously evaluated (and enriched) annotations. The following sections describe these steps in detail.

3.2 Representing Annotations in RDF

Annotations describing artworks provided by the users from the Web are represented using the Open Annotation Model [19] which helps to link annotations to the user who created them and the artifact for which an annotation was created. A subset of annotations are reviewed by the experts at the cultural heritage institutions and their reviews are represented as an annotation of an annotation. The review indicates an expert opinion about the annotation that the user provided according to standards of the institution. Apart from information about annotations, we would like to extend our information about the user who provided the annotation. For users who registered in the system and provided profile information, we model their information using the FOAF ontology [5], while anonymous users do not have any additional information in their profile. Also the artifact has some meta data such as the creator of the artifact, a title, and material properties. We use the Eurpeana Data Model (EDM) [10] to represent these properties. Fig. 3 shows our generic semantic model for the annotations contributed to the cultural heritage domain.

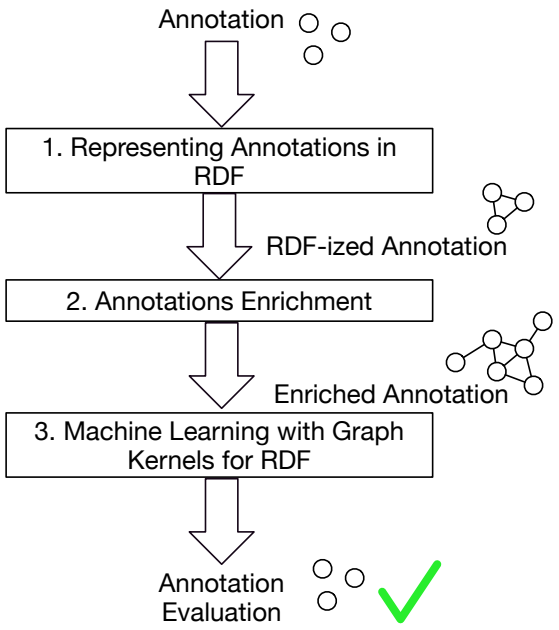

Fig. 2. Annotation evaluation workflow. First, the annotation is represented in RDF. Then it is enriched. Lastly, we use the RDF-based machine learning to predict its quality.

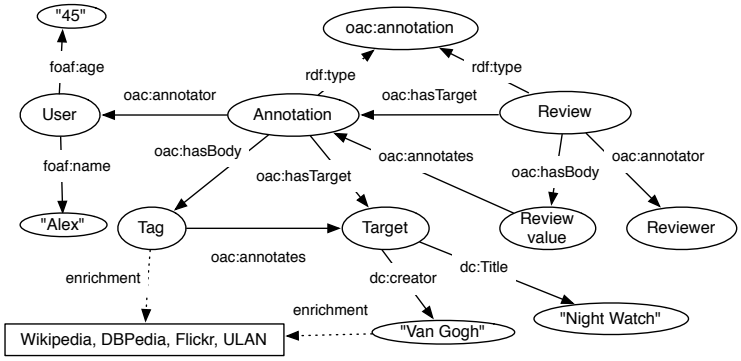

Fig. 3. Semantic model of Cultural Heritage annotations

3.3 Annotations Enrichment

Enrichment of the annotations is done since RDF graph kernels can easily use additional information since all additional information is part of RDF graph to make predictions. The properties related to the artwork, the creator of the artwork and the annotation itself are relevant to be enriched. Unfortunately, to the best of our knowledge, there were no publicly accessible knowledge repositories related to artworks. We extend the creator data using the Union List of Artist Names (ULAN) and DBPedia, and annotation data with DBPedia, Flickr and Wikipedia.

The ULAN is a structured vocabulary maintained by professionals of the Getty Research Institute and contains information such as date of birth and nationality of 202.720 past and current artists (in 2011[6]). Wikipedia[7] is a mostly unstructured knowledge base maintained by tens of thousands of volunteers worldwide and contains information on a very broad spectrum of topics. The information is intended for human consumption. DBpedia[8] is a semantic repository of information that is extracted from Wikipedia. Most pages on the English Wikipedia have a corresponding entry in DBPedia. Information in DBPedia is structured in RDF and is machine processable. Flickr is a website where people upload and share their images. Most images are tagged with descriptive labels.

Institutions store creator information either as structured, semi-structured or unstructured text. For linking purposes we assume creator text is unstructured. We map ULAN resources using the `getty:labelPreferred` (e.g. Rembrandt van Rijn) and `getty:labelNonPreferred`(e.g. Rembrandt Hermanszoon van Rijn) properties. We also map DBPedia resources of type `dbpedia-owl:Artist` using the `foaf:name` property.

The textual annotations are compared to DBPedia resources based on the `rdfs:label` property to check whether the annotation corresponds to existing words. The popularity of each annotation is calculated using Flickr by counting the number of images that have been uploaded in 2014 and were labeled with that annotation.

3.4 Machine Learning with Graph Kernels for RDF

In a typical machine learning classification task, one tries to predict a class for a set of instances. Each instance is represented by a feature vector: a list of properties of that instance. This approach fits well to the scenario where the dataset is a table in a database, and each instance is a row. But it does not easily translate to RDF graphs. For example, consider the simple RDF graph given in Fig. 4A. Suppose we want to predict a property of things (i.e. people) that are Persons, then our instances are the two nodes: person1 and person2. It is not immediately obvious what the features of person1 and person2 are.

[6] `http://www.getty.edu/research/tools/vocabularies/ulan/faq.html`
[7] `http://en.wikipedia.org/wiki/Wikipedia:About`
[8] `http://wiki.dbpedia.org/About`

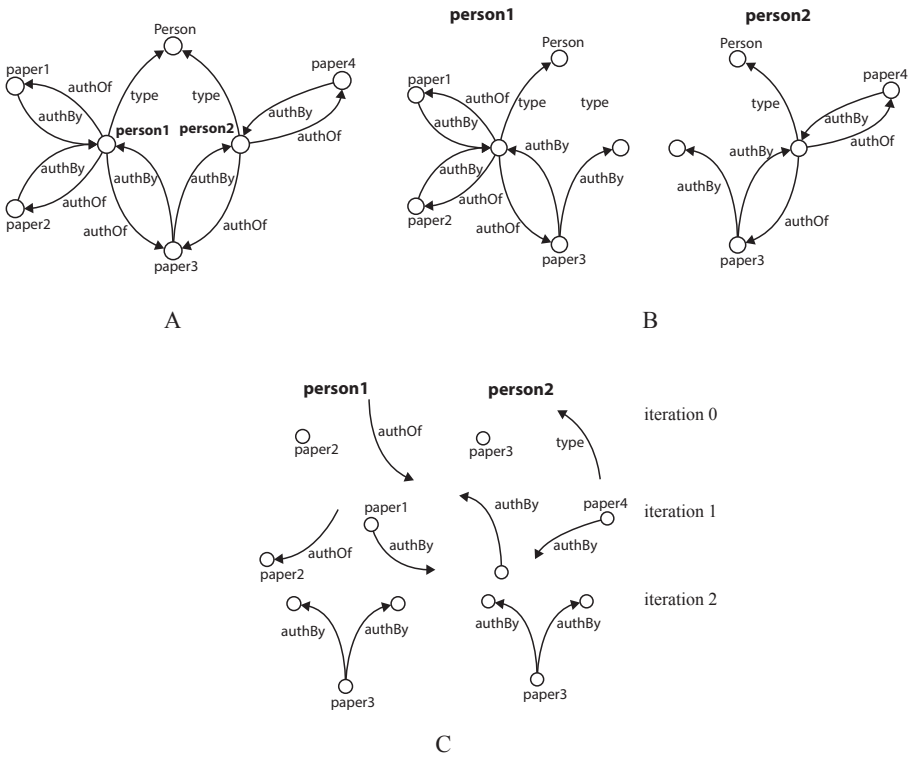

Fig. 4. Example RDF graph (A), with two subgraphs of depth 2 (B) and examples of extracted features (C).

Machine learning for RDF data using graph kernels was introduced in [13] as a way to deal with this issue by using structural patterns of the RDF graph as input for kernel based learning algorithms [20,21]. For each instance we consider the subgraph around that instance (up to a certain depth) as its 'features', see Fig. 4B. For these subgraphs structural properties are computed as something that is called a 'kernel', which is essentially a similarity function between objects, for instance, between subgraphs of an RDF graph. This kernel is used as the input data for a learning algorithm. The main advantage of using graph kernels for learning from RDF, compared to other techniques, is that it is a generically applicable and flexible approach [16]. Little knowledge of the dataset is required to use these methods and it allows for easy integration of additional knowledge into the learning process, by simply adding triples to the RDF graph.

In this paper we will use the Weisfeiler-Lehman [22] graph kernel for RDF (WLRDF), introduced in [6]. This is a state of the art graph kernel for learning from RDF data in terms of prediction accuracy, with very good computational performance. For each instance, the WLRDF kernel efficiently computes subtree patterns as features, in a number of iterations, where each iteration computes more complex patterns. These patterns are illustrated in Fig. 4C. Typically, the features that are considered by a kernel are computed implicitly. However, subtree features of the WLRDF kernel are computed explicitly and we can therefore inspect which subtree patterns are important in the learning process.

As our learning algorithm, we use the well-known Support Vector Machine (SVM). SVMs are very efficient and robust classification algorithms that try to separate classes by finding a maximally separating hyperplane. For more see for example the books [20,21].

In the machine learning step in our workflow, the instances that we use are annotations, i.e. nodes that are of type Annotation. For each annotation a subgraph is extracted up to a specified depth. From Fig. 3 we can see that larger depths leads to the inclusion of more levels of information in the graph. The WLRDF kernel is computed using these subgraphs and then used to train a SVM on labelled (in terms of quality) annotations. This SVM is then used to predict the annotation quality of unseen annotations.

4 Experimental Setup

We apply our approach on the Steve.museum dataset, which is described in Section 4.1. The details of the enrichment step is discussed in 4.2 followed by the experimental parameters set to run the experiment in Section 4.3.

4.1 Steve.museum Dataset

The Steve.museum [12] project was started together with United States art museums with the aim to explore the role that user-contributed descriptions can play in improving on-line access to works of art. Annotations were gathered for

1,784 artworks and the usefulness of each annotation was evaluated by professional museum staff. The reviewers distinguished 12 categories of usefulness. The category `usefulness-useful` and `usefulness-not_useful` indicated positive and negative usefulness. Other categories described why the annotation was not useful (e.g. `problematic-misspelling`, `judgement-positive`). The annotations including their evaluations and annotator information were published as a SQL dataset.

The dataset contains 49,767 artifacts annotations in total, along with the related metadata, created by 730 anonymous[9] and 488 registered users, where anonymous users created 24.016 annotations (43% of the total). Registered users could enter additional profile information. Table 1 lists those properties and the percentage of registered users who provided a value for a property.

There are some differences in the behaviour of anonymous and registered users: the first contributed on average 15 annotations per session, the latter 33. Moreover, we see a clear pattern in the week day distribution: registered users contribute annotations mostly between Tuesday and Thursday and the anonymous users during the other days of the week. Also, registered users contributed most of their annotations in the morning and in the evening, although the pattern here is less definite.

Out of the 49,767 annotations, 48,789 (98%) have been evaluated, of which 87% as `usefulness-useful`. Table 3 shows the average performance per session of the registered and anonymous users. The annotations contributed by the registered users are of slightly higher quality than those contributed by anonymous users.

4.2 Dataset Transformation

We transformed the data into Linked Data using the model illustrated in Figure 3. Most properties of the users and the annotation could be mapped one-to-one. However, some annotations were reviewed multiple times. For the purpose of prediction we required each annotation to have exactly one review; therefore, we applied the following strategy: if any of the reviews stated the usefulness of an annotation as `usefulness-useful`, we selected that review, giving more weight to a potentially useful annotation. If not, we selected the usefulness value with the single highest frequency. When there were multiple reviews with the highest frequency, we removed the annotation as this happened in very few cases. This resulted in the deletion of 1,246 annotations leaving 47,543 annotations. Also we removed the reviewer information from the graph since that information would not be present for future (un-reviewed) annotations which we want to automatically assess.

[9] Anonymous users were identified using a disambiguation process based on their web session identifier since multiple annotations may have been created by the same user at different times. However, we do not know if two sessions were created by the same anonymous user, but for registered users we see that this happens quite rarely: the average number of sessions per registered user is 1.03.

The Steve.museum dataset contains 1,082 unstructured creator names. Our goal was to identify creators pointing to individual persons. Therefore we filtered the creator names containing the string *unknown*, locations (countries and places), time periods, and hashed strings to anonymize the details of certain artefacts. This resulted in 742 creator strings (of which some could still point to the same person) which we considered candidate artists. When possible we put the name in $< firstname >< lastname >$ order. We used the preprocessed name to match to DBPedia and ULAN.

For each name that could not be matched we performed a Wikipedia search on that name where we automatically retrieved the top 5 results and checked if the corresponding DBPedia resources were of the type `dbpedia-owl:Artist`. We automatically made the mapping if there was only one Artist in the results and decided manually when there were multiple Artists. In total 579 candidate artists were mapped onto 479 distinct DBPedia resources. For the ULAN mapping we used both the preprocessed name and the spelling variations on DBPedia if there was a match. In total 470 candidate artists were mapped to 422 distinct ULAN resources. The mapping process resulted in 605 mapped candidates of which 442 mapped by both ULAN and DBPedia, 138 only mapped to DBPedia and 27 only mapped to ULAN.

To enrich the annotation as described in Section 3 we tokenized the annotation and removed stopwords, special characters such as "" and ">", and words of length 1. We added a `custom:wikipediaMatchCount` property to each annotation with the number of matched words from the preprocessed annotation. For Flickr we used the `flickr.photos.search` API function searching for all photos containing all annotation words as label and which were uploaded in 2014. We added a `custom:flickrMatchCount` property to each annotation with the amount of photos returned by the API. Finally, to match with the Wikipedia description of the creators we tokenized and stemmed the description, stemmed the preprocessed annotation words and added a `custom:hasCreatorMatchCount` property indicating the amount of matched words.

Table 2 provides a summary of the complete dataset. The transformed dataset and the enrichments are available as RDF/XML files online.[10]

Community	**Experience**	**Education**	**Age**	**Gender**	**Household income**
431 (88%)	483 (99%)	483 (99%)	480 (98%)	447 (92%)	344 (70%)
Works in a museum	**Involvement level**	**Tagging experience**	**Internet connection**	**Internet usage**	
428 (88%)	411 (84%)	425 (87%)	406 (83%)	432 (89%)	

Table 1. Annotator properties and the percentage of registered annotators who filled in the property.

[10] The dataset can be downloaded at `https://www.dropbox.com/s/0l8zo023hhsrsjt/all_data.zip?dl=0`.

Total number of triples	473,986
Annotators / registered annotators	1,218 / 488 (40%)
Annotated artworks	1,784
Annotations / unique annotations	45,733 / 13,949 (31%)
Candidate creators / mapped creators	1,082 / 605 (56%)
Annotations in Flickr (> 0 images retrieved)	25,591 (56%)
Annotations in DBpedia (> 0 words matched)	25,163 (55%)

Table 2. Summary of the transformed and enriched Steve.museum dataset.

Evaluation Category	**Average frequency per session (Registered users)**	**Average frequency per session (Anonymous users)**
usefulness-useful	75.57%	74.46%
usefulness-not_useful	11.19%	11.96%
problematic-personal	0.53%	0.61%
problematic-no_consensus	0.69%	0.63%
problematic-foreign	0.99%	1.13%
problematic-huh	0.36%	0.55%
problematic-misperception	2.65%	2.76%
problematic-misspelling	0.88%	0.89%
judgement-positive	0.70%	0.48%
judgement-negative	0.75%	0.95%
comments	2.15%	1.72%
not evaluated	3.54%	3.86%

Table 3. Comparison of the average performance per session between registered and anonymous users.

4.3 Experimental Parameters

As can be seen in Table 3 the distribution of the usefulness categories is very skewed and many categories are very small. For our experiments we therefore kept the larger `usefulness-useful` and `usefulness-not_useful` categories, grouped together both `problematic` and `judgement` subcategories and removed both the `comments` category and annotations which were not evaluated.

Our experiments have been implemented in Java using the 'mustard' library[11], which implements different graph kernels, such as the WL RDF kernel, for RDF data and wraps the Java versions of the LibSVM [4] and LibLINEAR[12] [8] SVM libraries.

[11] Our code can be found in the `org.data2semantics.mustard.experiments.IFIPTM` package of the library at `https://github.com/Data2Semantics/mustard`

[12] `http://liblinear.bwaldvogel.de/`

The experiments were run on depth 1 (including annotation properties), depth 2 (additionally including annotator and artwork properties) and depth 3 (additionally including properties from the linked datasets). On each depth we created 10 subsets of the graph and performed a 5-fold cross-validation, optimizing the C-parameter of the SVM in each fold, using again 5-fold cross-validation. The number of iterations parameter h for the WLRDF kernel was fixed to the depth $\times 2$. This parameter can also be optimized, however this has relatively little impact, since the higher iterations include the lower iterations. Subsets were created by taking a random sample of annotations in the `usefulness-useful` category of size equal to the other categories combined and took all annotations from those categories. Each subset contained approximately 9000 annotations. For each depth and subset we calculated the accuracy, precision, recall and F1 score for the categories combined and individually.

5 Results and Analysis

In this section we present our experimental results. First we give our quantitative results and then we qualitatively analyse important features for predicting the different categories.

5.1 Comparison of Accuracy, Precision and Recall for Predictions at Different Depths

We compare the accuracy, F1-measure, precision and recall for predicting four different categories (`usefulness-useful`, `usefulness-not_useful`, `judgment`, `problematic`) at three different depths of the graph and present the results in Table 4. The features for the graph which were included at different depths are described in Section 4.3. We repeated the experiment for predicting two types of review categories (`usefulness-useful` and `usefulness-not_useful`) and found that the results are comparable to the ones mentioned in Table 4, while the overall F1-measure was higher, with 0.76 for every depth. This is to be expected since the two classes which were hard to predict were not included. The best overall results were achieved under the depth 2 setting. The `judgement` class is very hard to predict, as we can see from the very low precision, recall and f1 scores.

5.2 Comparison of Relevant Graph Features at Different Depths

The multi-class SVM implementation in LibLINEAR computes a SVM for each class, which can be used to identify the important graph features for each class. Thus, we trained a SVM for the first of our 10 four-class subsets. A manual analysis of these important features (those with the highest weight) for the different classes at different depths shows some interesting results. We will not mention the results for the `judgement` class, since it was predicted very poorly.

Depth	Prediction class	Avg. Accuracy	Precision	Recall	F1 measure
1	Usefulness-useful		0.75	0.78	0.76
	Usefulness-not_useful		0.74	0.74	0.74
	Judgement		0.00	0.00	0.00
	Problematic		0.68	0.25	0.37
	All classes	0.75	0.54	0.44	0.47
2	Usefulness-useful		0.77	0.77	0.77
	Usefulness-not_useful		0.74	0.75	0.75
	Judgement		0.30	0.04	0.07
	Problematic		0.64	0.34	0.45
	All classes	0.75	0.61	0.48	0.51
3	Usefulness-useful		0.77	0.76	0.77
	Usefulness-not_useful		0.74	0.76	0.75
	Judgement		0.05	0.01	0.01
	Problematic		0.64	0.32	0.42
	All classes	0.75	0.55	0.46	0.49

Table 4. Comparison of Results from Predictions Using the WLRDF Kernel at Different Depths

At depth 1, the `useful` class has a large number of specific date strings, e.g. "2007-07-18T00:22:04", as important features. However, the `not-useful` class is recognized by features pointing to the artwork that is annotated, such as `oac:hasTarget->http://purl.org/artwork/1043`. The `problematic` class has important features similar to the `useful` class.

Graph features containing `edm:object_type` and `oac:hasBody` are almost exclusively the most important features for identifying `useful` annotations at depth 2 and 3. In contrast, the type of features that are used in classifying `not-useful` annotations is more diverse. They include graph features with the material used in the artwork or information about the annotators. For example a set of important features has the graph pattern that includes the information that the annotator has "Intermediate" experience. The `problematic` class at depth 2 and 3 is recognized with very specific features, like date strings, that are not as general as for the other two classes.

6 Discussion and Future Work

In this paper we presented a workflow to convert datasets in the Cultural Heritage domain to RDF and to enrich the datasets to be used for predictions of annotation quality using RDF graph kernels. We have provided both a qualitative and quantitative analysis of the results and have shown that RDF kernels are quite beneficial in making predictions about quality.

From our experiments it can be seen that employing RDF graph kernels helps in predicting classes of annotations with a overall best accuracy of 75%, which is a good rate of acceptance. The single class measures of accuracy, precision, recall and f1-measure for the classes of `judgement` and `problematic` are not

useful since the percentage of their classes were too small to perform a good training and thus they were predicted badly.

We also identified which features are relevant at different depths to make the predictions per category and provided an analysis. The features which are relevant to predict a certain class of quality are useful to design annotation tasks in the future. If a particular creator is selected as a relevant feature and if the majority of annotations by different users to an artwork belonging to that creator tend to be evaluated mostly as `usefulness-not_useful`, then it might indicate that the annotation task is difficult for that particular artwork. Similarily for different datasets such in-depth analysis helps to re-design the annotation tasks to obtain better quality from the crowds.

The approach of using graph kernels for RDF is very flexible as additional information can easily be added to the learning process by extending the RDF graph. However in Steve.museum dataset, some node labels provide very specific information, which is not beneficial for generalization. For example, the annotations are timestamped with exact times in seconds, whereas the day of the week might be more informative. Some (light) graph pre-processing can help to alleviate these issues, without hindering the flexibility and extensibility of the approach. We will investigate this in future work.

The automatic prediction of quality of annotations based on their metadata helps Cultural Heritage institutions alleviate the task of reviewing large number of annotations and helps to add the most useful annotations directly to their system for better search and retrieval through their collection. As part of future work, we would like to perform our experiments on different datasets from the Cultural Heritage domain to understand how and which features are most relevant in predicting quality from these datasets.

Acknowledgement This publication is supported by the Dutch national program COMMIT.

References

1. D. Artz and Y. Gil. A survey of trust in computer science and the semantic web. *Journal of Semantic Web*, 2007.
2. D. Ceolin, A. Nottamkandath, and W. Fokkink. Automated evaluation of annotators for museum collections using subjective logic. In *IFIPTM*, pages 232–239. Springer, 2012.
3. D. Ceolin, A. Nottamkandath, and W. Fokkink. Efficient semi-automated assessment of annotation trustworthiness. *Journal of Trust Management*, 1:1–31, 2014.
4. C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.
5. L. M. Dan Brickley. FOAF. `http://xmlns.com/foaf/spec/`, Jan. 2014.
6. G. K. D. de Vries. A fast approximation of the Weisfeiler-Lehman graph kernel for RDF data. In H. Blockeel, K. Kersting, S. Nijssen, and F. Zelezný, editors, *ECML/PKDD (1)*, volume 8188 of *Lecture Notes in Computer Science*, pages 606–621. Springer, 2013.

7. A. Doan, R. Ramakrishnan, and A. Y. Halevy. Crowdsourcing systems on the world-wide web. *Commun. ACM*, 54(4):86–96, Apr. 2011.
8. R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.
9. J. Golbeck. Trust on the World Wide Web: A Survey. *Foundations and Trends in Web Science*, 1(2):131–197, 2006.
10. S. Hennicke, M. Olensky, V. de Boer, A. Isaac, and J. Wielemaker. *A data model for cross-domain data representation. The "Europeana Data Model" in the case of archival and museum data.* 2011.
11. O. Inel, K. Khamkham, T. Cristea, A. Dumitrache, A. Rutjes, J. van der Ploeg, L. Romaszko, L. Aroyo, and R.-J. Sips. Crowdtruth: Machine-human computation framework for harnessing disagreement in gathering annotated data. In *ISWC 2014*, volume 8797 of *Lecture Notes in Computer Science*, pages 486–504. Springer, 2014.
12. U. institute of Museum and L. Services. Steve Social Tagging Project, Jan. 2012.
13. U. Lösch, S. Bloehdorn, and A. Rettinger. Graph kernels for RDF data. In *ESWC*, pages 134–148, 2012.
14. A. Nottamkandath, J. Oosterman, D. Ceolin, and W. Fokkink. Automated evaluation of crowdsourced annotations in the cultural heritage domain. In *URSW*, volume 1259 of *CEUR Workshop Proceedings*, pages 25–36. CEUR-WS.org, 2014.
15. T. K. Prasad, P. Anantharam, C. A. Henson, and A. P. Sheth. Comparative trust management with applications: Bayesian approaches emphasis. *Future Generation Computer Systems*, pages 182–199, 2014.
16. A. Rettinger, U. Lösch, V. Tresp, C. d'Amato, and N. Fanizzi. Mining the semantic web—statistical learning for next generation knowledge bases. *Data Min. Knowl. Discov.*, 24(3):613–662, 2012.
17. M. Ridge. Introduction. In M. Ridge, editor, *Crowdsourcing Our Cultural Heritage*, Digital Research in the Arts and Humanities. Ashgate, Farnham, October 2014.
18. J. Sabater and C. Sierra. Review on Computational Trust and Reputation Models. *Artificial Intelligence Review*, 24:33–60, Sept. 2005.
19. R. Sanderson, P. Ciccarese, H. V. de Sompel, T. Clark, T. Cole, J. Hunter, and N. Fraistat. Open annotation core data model. Technical report, W3C Community, May 9 2012.
20. B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, 2001.
21. J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, 2004.
22. N. Shervashidze, P. Schweitzer, E. J. van Leeuwen, K. Mehlhorn, and K. M. Borgwardt. Weisfeiler-Lehman graph kernels. *J. Mach. Learn. Res.*, 12:2539–2561, Nov. 2011.
23. P. Singh, T. Lin, E. T. Mueller, G. Lim, T. Perkins, and W. L. Zhu. Open mind common sense: Knowledge acquisition from the general public. In *DOA, CoopIS and ODBASE 2002*, pages 1223–1237, London, UK, UK, 2002. Springer-Verlag.
24. R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng. Cheap and fast—but is it good?: Evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 254–263. Association for Computational Linguistics, 2008.
25. L. von Ahn and L. Dabbish. Labeling images with a computer game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pages 319–326. ACM, 2004.