Learning to Cite: Transfer Learning for Digital Archives

Dennis Dosso[1][0000−0001−7307−4607], Guido Setti[2], and Gianmaria Silvello[1][0000−0003−4970−4554]

[1] Department of Information Engineering, University of Padua
{dosso, silvello}@dei.unipd.it
[2] Department of Mathematics, University of Padua

Abstract. We consider the problem of automatically creating citations for digital archives. We focus on the learning to cite framework that allows us to create citations without users or experts in the loop. In this work, we study the possibility of learning a citation model on one archive and then applying the model to another archive that has never been seen before by the system.

1 Introduction

Scientific research relies more and more on data for conducting advanced analysis and support new discoveries and empirical findings. Nowadays, scientific datasets constitute the backbone of the system of the sciences and are key factors for conducting high-quality research. Hence, open and shared datasets constitute first-class objects of the scientific process and need to be retrieved, accessed and cited as traditional scientific articles are. Moreover, the creation, curation, and preservation of scientific datasets require a great deal of investment and human-effort that need to be recognized and assessed by the scientific community [9].

For these reasons, there is a strong demand [1, 5] to give databases the same scholarly status of traditional references. Recently, data citation has been defined as a computation problem [3], where the main issues to be tackled are: (i) the unique and persistent identification of a dataset or a data subset; (ii) the temporal persistence of the data as well as of the data citations; and, (iii) the automatic creation of text snippets (references) citing data subsets.

In this work, we focus on the automatic creation of text snippets for citing datasets with a variable granularity. This problem has been tackled with rule-based/deterministic approaches from a relational [11], hierarchical [4] and graph [2, 7] database perspective or with a machine learning approach – i.e. the *Learning to Cite* (LtC) approach – for XML [8]. Both approaches present their pros and cons. The LtC approach has the main advantage of not requiring expert intervention to define citation policies and rules required by rule-based systems; on the other hand, the citations produced are not "exact" as those produced by rule-based systems.

In [8], we introduced the LtC approach and we tested it on digital archives (i.e. XML Encoded Archival Description (EAD) files). We decided to testbed

the LtC approach on this domain because archives usually lack resources and present a high data heterogeneity also within the same archive; these aspects make the LtC approach particularly valuable in the archival context. In [8] we showed that the LtC approach allows us to produce quite accurate citations with small training sets, thus requiring minimum effort to database administrators and domain experts.

In this work, we move one step ahead by studying if *"it is possible to learn a citation model on an archive A and apply it to an unseen archive B"*. The goal is to learn a citation model on an archive where there are enough economic and human resources to build a training set and maintain a citation model (e.g. the LoC archive) and to apply it to other – possibly a broad spectrum – archives with lower resources.

Hence, in this work, we study the problem of transfer learning from an archive to another by using the LtC approach presented in [8] as a baseline. We conduct two experiments: (i) we train a citation model on a uniform and consistent training set (i.e. EAD files coming from a single archive) and we create citations for EAD files coming from five different heterogeneous archives; and, (ii) we train a citation model on a training set composed of the union of five heterogeneous archives and we create citations for a single archive not present in the training set. These two tests define a task harder than a traditional transfer learning task because the training and the test sets we consider are not only disjointed, but they belong to different datasets altogether. We show that the LtC approach has the potential to be applied in a transfer learning scenario, even though there is a performance drop with respect to a classical learning scenario where training and test sets are sampled from the same archival collection.

The rest of paper is organized as follows: in Section 2 we briefly describe how data citation applies to the archival domain and summarize the main approaches to data citation. In Section 3 we describe at a high-level the LtC approach and explain how we model transfer learning for data citation. In Section 4 we define the experimental setup, describe the datasets and the experiments we conduct and in Section 5 we present the results of the evaluation. Finally, in Section 6 we draw some conclusions and outline future work.

2 Related Work

2.1 Digital Archives

Archives are composed of unique records where the original order of the documents is preserved because the context and the order in which the documents are held are as valuable as their content. Archival documents are interlinked and their relationships are required to understand their informative content. Therefore, archives explicitly model and preserve the provenance of their records by means of a hierarchical method, which maintains the context in which they have been created and their relationships.

Archival descriptions are encoded by means of the Encoded Archival Description (EAD) which is an XML description of a whole archive; EAD files resemble

the description of the archival material and provide a means to represent the internal logic of an archive.

The EAD files represent a good test-bed for the LtC approach because they are deep files not easy to navigate and understand for the users, there is a wide variability in the use of tags that makes it difficult to set up citation rules across files and every node in an EAD file is a potential citable unit.

2.2 Data Citation Approaches

A recent and detailed overview of the theory and practice of data citation can be found in [9]. It has been highlighted that the manual creation of citation snippets is a barrier towards an effective and pervasive data citation practice as well as a source of inconsistencies and fragmentation in the citations [10]. Indeed, especially for big, complex and evolving datasets, users may not have the necessary knowledge to create complete and consistent snippets.

Recently, some solutions to tackle the problem of automatically creating citation snippets have been proposed. There are two main approaches: (i) rule-based and, (ii) machine-learning based.

Within the first approach, one of the first methods has been proposed by [4]. This method requires that the nodes corresponding to citable units are identified and tagged with a rule that is then used to generate a citation. This method was extended by [3] which defined a view-based citation method for hierarchical data. The idea is to define logical views over an XML dataset, where each view is associated with a citation rule, which if evaluated generates the required citation snippet according to a predefined style. This approach has been further formalized and extended also for the relational databases in [11]. In the same vein by exploiting database views, [2] proposed a system for citing single RDF resources by using a dataset on the medical domain as use-case.

The machine learning-based approach has been proposed for the first time by [8] with the Learning to Cite (LtC) framework that we present below.

3 Creation of Data Citations Based on Machine Learning

3.1 Learning to Cite

The aim of the LtC approach is to automatically create a model that can produce human- and machine-readable citation from XML files (EAD files for our use-case) without manual interventions of the data curators and without any modification to the data to be cited.

The LtC framework is composed of six main blocks as shown in Figure 1: the training data, the learner, the citation model, the citation systems, the test data and the output reference.

The training set is composed of a collection $\mathcal{C}$ of XML files. Given two sets $T = \{t_1, t_2, \ldots, t_n\}$ of XML trees and a set $\mathcal{H} = \{H_1, H_2, \ldots, H_m\}$ of human-readable citations, the *learner* component takes as training data a set of pairs

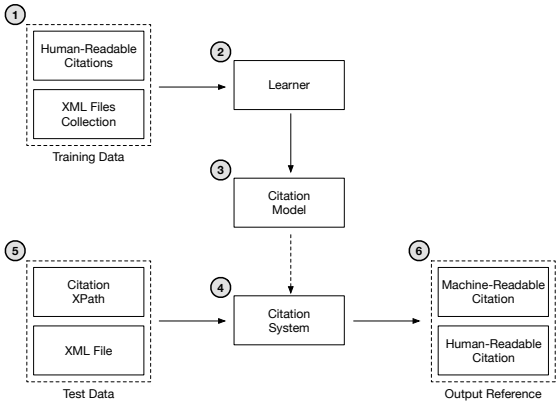


Fig. 1: The building blocks of the "Learning to Cite" framework [8].

$< t_i, H_i >$. In particular, each citation $H_i \in \mathcal{H}$ is associated to one and only one XML tree $t_i \in T$, while each tree has at least one associated citation (but potentially more than one).

From the training data the *learner* produces a *citation model* able to create human-readable citations. In particular, the test data are a set of pairs $< p_t, t_t >$ where t_t is a XML tree with a citable unit referenced by the XPath p_t. The *citation system* parses the XPath p_t and creates a human-readable citation for the user exploiting the data inside the XML.

In particular, for the validation phase, the system can use a function to evaluate the effectiveness of the citation system and tune the parameters. These functions are *precision*, *recall* and *fscore*. Let $MC_k = \{p_1, p_2, \ldots, p_n\}$ be a machine readable citation generated by the system for the element $e_k =< p_k, t_k >$. $\{p_1, p_2, \ldots, p_n\}$ are the paths composing the citation. Let $GTC_k = \{p_1^{'}, p_2^{'}, \ldots, p_m^{'}\}$ be the ground-truth machine-readable citation for the same element, i.e. the set of paths that correctly form the citation for e_k. Then we can define:

$$precision = \frac{|MC_k \cap GTC_k|}{|MC_k|}$$

$$recall = \frac{|MC_k \cap GTC_k|}{|GTC_k|}$$

$$f\text{-}score = 2 * \frac{precision * recall}{precision + recall}$$

Precision is the ratio between the total number of correct paths in the generated citation with the total number of generated paths, while recall is the ratio between the total number of generated correct paths and the total number of correct paths. Both are in the $[0, 1]$ interval, just like the fscore, which is a synthesis measure.

The framework uses one of these function in a k-fold validation strategy to find the best parameters for the system.

3.2 Transfer Learning

As defined in [6], given a source domain $\mathcal{D}_S$ with a learning task $\mathcal{T}_S$ and a target domain $\mathcal{D}_T$ with a learning task $\mathcal{T}_T$, *transfer learning* aims to help improve the learning of the target predictive function $f_T(\cdot)$ in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$ or $\mathcal{T}_S \neq \mathcal{T}_T$.

In order to apply transfer learning to the LtC approach, it is necessary to define the domains and task at hand:

- The *source domain* $\mathcal{D}_S$ is the couple $\{\mathcal{C}_S, P(X_S)\}$, where $\mathcal{C}_S$ is a collection of XML files, X_S is a sub-collection sampled from $\mathcal{C}_S$ and P is a marginal probability distribution on X_S;
- The *source task* $\mathcal{T}_S$ is the couple $\{\mathcal{Y}_S, f_S(\cdot)\}$, where $\mathcal{Y}_S$ is the set of ground truth machine-readable citations for $\mathcal{C}_S$, and $f_S(\cdot)$ is the function represented by the model built from the training data obtained from $\mathcal{C}_S$;
- The *target domain* $\mathcal{D}_T$ is the couple $\{\mathcal{C}_T, P(X_T)\}$, where $\mathcal{C}_T$ is a different collection of XML files, X_T is a sub-collection sampled from $\mathcal{C}_T$ and P is a marginal probability distribution on X_T.
- The *target task* $\mathcal{T}_T$ is a couple $\{\mathcal{Y}_T, f_T(\cdot)\}$, where $\mathcal{Y}_T$ is the set of ground truth machine-readable citations for $\mathcal{C}_T$, and $f_T(\cdot)$ is the model build from the training data of $\mathcal{C}_T$.

Thus, we can use the knowledge coming from the source domain and source task to learn the predictive function $f_T(\cdot)$, which corresponds to a citation model for the target collection. In the case we are considering, source and target domain are different, and so are source and target tasks.

4 Experimental Setup

4.1 Experimental Collections

The first experimental collection we consider is based on the Library of Congress (LoC)[3] EAD files and it has been defined in [8]; it consists of training, validation and test set. The training and validation sets are composed of XML tree and human-readable citation pairs. The validation set is obtained with k-fold cross validation from the training set. The test set is made of XML tree and machine-readable citation pairs. The human- and machine-readable citations were all built manually. The full LoC collection is composed of 2,083 files. In order to build the training and validation set, 25 EAD files were randomly selected, and from each of these files 4 citable units were extracted. For each citable unit, a human-readable and machine-readable citation was manually created to be used to train the citation system and to build the ground-truth to be used for validation purposes respectively. The test set was built by following a similar procedure. In this case, a ground-truth machine-readable citation was manually built for every randomly sampled citable unit. A new collection of EAD files

[3] http://findingaids.loc.gov

is created in order to test Transfer Learning. Five different and heterogeneous source archives are selected:

1. University of Chicago Library finding aids (chicago);
2. University of Maryland Libraries finding aids (MdU);
3. Nationaal Archief, Den Haag (NL-HaNA);
4. Syracuse University finding aids (syracuse);
5. WorldCat aggregate collection aids (worldcat). This is a very heterogeneous collection of digital archives from all across the United States.

10 citable units are randomly selected from each of these collections, creating a new collection of 50 citable units from different sources. This new collection is called ***EAD various*** in the rest of the paper.

We conduct two experiments. In the first one, the source collection $\mathcal{C}_S$ is the LoC collection and the target task is to produce a set of citations for the target collection $\mathcal{C}_T$ which is the EAD various collection. The EAD various collection is only used as test set, hence a model $f_T(\cdot)$ cannot be directly built. $f_S(\cdot)$ will be used in order to learn the target predictive function $f_T(\cdot)$ for the target task $\mathcal{T}_T$. The second experiment reverses experiment one since we train on EAD various and test on LoC.

For each experiment, the citation model is built using the training and validation sets of the source collection and a 5-fold cross-validation is used to choose the best parameters of the model, with the f-score as optimization measure. The whole training and test procedure are repeated with different training sizes – i.e. the number of citable units contained in the training set – ranging from 20 to 80 with step 10. The citation model is then tested against the target collection and the procedure is repeated 5 times for each training size. The final measures presented are the average over the results of the five repetitions.

5 Evaluation

In the first experiment, we trained a citation model on the uniform LoC collection and we tested both on the LoC (classic learning procedure) and the EAD-various (transfer learning) test set.

In Table 1 we can see the precision, recall, and f-score obtained by the LtC approach with different training set sizes over the two considered test collections. As expected, the citation model produces generally better citations for the LoC collection, while for the EAD various collection the performances are usually halved. It is particularly interesting how a training set size of 20 immediately obtains acceptable values in all three measures. This is true for both the collections. This is consistent with the results presented in [8].

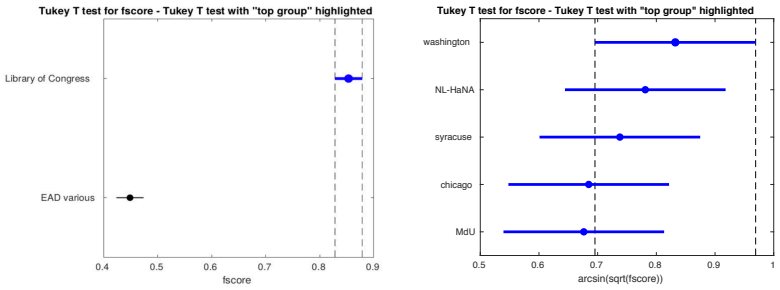
We conduct an ANOVA statistical test to check if the performance difference between LoC and EAD various are statistically significant. Figure 2a shows that the difference between the evaluation measure of the LoC collection and the EAD various collection are statistically significant and not due to chance. This

Table 1: Experiment 1: From homogeneous to heterogeneous. Precision, recall and f-score values obtained in the two test collections with f-score as optimization measure.

	Precision		Recall		f-score	
training set size	LoC	EAD various	LoC	EAD various	LoC	EAD various
20	0.8819	0.5289	0.8232	0.4188	0.8441	0.4584
30	0.9034	0.5283	0.8089	0.4101	0.8465	0.4525
40	0.8748	0.5282	**0.8312**	**0.4254**	0.8447	**0.4623**
50	0.9239	0.5277	0.8045	0.4043	**0.8531**	0.4488
60	**0.9333**	**0.5414**	0.7723	0.3955	0.8366	0.4474
70	0.9012	0.5284	0.8106	0.4131	0.8462	0.4547
80	0.9152	0.5281	0.8055	0.4065	0.8506	0.4503

means that the citation model built using the LoC is missing some knowledge regarding the target EAD various collection.

Given that the performances of the citation model are worse for the EAD various collection, we performed the Tukey's HSD test to check if the model is statistically different over the 5 subsets of citation units comprising the *EAD various* collection. The results, presented in Figure 2b, shows that the citation model built with the LoC collection training data (using 50 citation units) behaves with no significant difference with regard to f-score on the 5 sub-collections (the same result is obtained with precision and recall).



(a) fscore of LoC vs EAD various (b) fscore of single subcollections

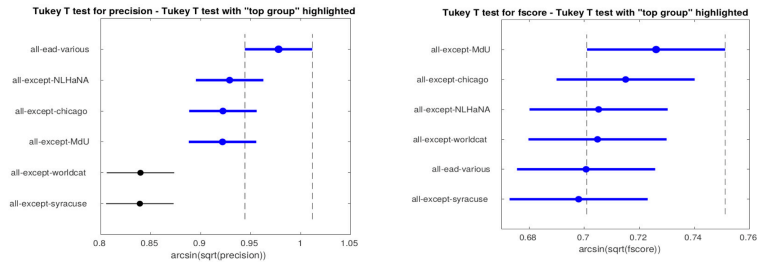
Fig. 2: (a) The Tukey's HSD test for fscore for the different sub-collections of *EAD various*. Training size is 50. (b) .

The second experiment builds the citation model with the *EAD various* collection and tests it on the LoC collection. The aim of this experiment is to discover if one of the five sub-collections of *EAD various* is more informative than the other in an LtC setting. The citation model has been built six times. One using all *EAD various* as training set, and the remaining five times by leaving out of the training set one sub-collection at a time.

Table 2: Precision, recall, and f-score values obtained from 5 different citing models trained on the *EAD various* collection, leaving out each time one of the sub-collections. f-score is the optimization measure. The training set is 40.

collection left out	precision	recall	fscore
none	0.4143	**0.4531**	0.4199
chicago	0.4366	0.4498	0.4334
MdU	**0.4577**	0.4518	**0.4439**
NL-HaNA	0.4191	0.4530	0.4236
syracuse	0.4144	0.4430	0.4178
worldcat	0.4204	0.4485	0.4238

Table 2 shows that the performances obtained with the full EAD various collection are comparable to those obtained by leaving out one sub-collection. These are probably due to the heterogeneity of the training collection and to the different employment of tags among the sub-collections with respect to the LoC collection. Moreover, as shown in Figure 3b the differences between the sub-collections are not statistically significant.



(a) precision (b) fscore

Fig. 3: The Tukey's HSD tests for the measures of precision and fscore of the models trained on the whole *EAD various* collection (all-ead-various) and on different sub-collections, obtained with the leave-one-out method, and tested on the LoC collection. Training size of 40 (except for the all-ead-various).

Nevertheless, let's note that there the sub-collection worldcat and syracuse appear to be determinant for the performances of the model in terms of precision (Figure 3a). In fact, when we remove them from the training set, the performances are significantly lower. Also, the recall measure doesn't highlight any significant difference (thus omitted from the plots).

6 Final Remarks

The transfer learning experiments we conducted highlight that different digital archives employ the EAD standard differently and use the tags in a heterogeneous way. This aspect impacts the LtC approach which behaves very well within the same archive, but fairly less when we try to apply the same model on a heterogeneous collection.

We see that the impact of a single sub-collection in the training set is marginal even though some collections bring key contributions that help to improve the citation model – see the role of the highly heterogeneous "worldcat" sub-collection in the second experiment. From the experiments, we see that the LtC framework performs at best when trained on the collection where it will be applied. On the other hand, the framework adapts well to heterogeneous collections, provided that the test set is not composed of EAD files coming from archives not considered in the training set. Finally, we confirm that the LtC approach does not require big training sets as it was shown for a homogeneous setting in [8].

Future work will investigate the role of expert users in a reinforcement learning setting, where the citations produced by the system are corrected and revised by experts. We plan to dynamically change the citation model when a user modifies a citation in order to add a new learning layer to the system.

Acknowledgments

This work is supported by the Computational Data Citation (CDC-STARS) project financed by the University of Padua.

References

1. Out of Cite, Out of Mind: The Current State of Practice, Policy, and Technology for the Citation of Data, vol. 12. CODATA-ICSTI Task Group on Data Citation Standards and Practices (September 2013)
2. Alawini, A., Chen, L., Davidson, S.B., Portilho Da Silva, N., Silvello, G.: Automating Data Citation: The eagle-i Experience. In: 2017 ACM/IEEE Joint Conference on Digital Libraries, JCDL 2017. pp. 169–178. IEEE Computer Society (2017)
3. Buneman, P., Davidson, S.B., Frew, J.: Why data citation is a computational problem. Communications of the ACM (CACM) **59**(9), 50–57 (2016)
4. Buneman, P., Silvello, G.: A Rule-Based Citation System for Structured and Evolving Datasets. IEEE Data Eng. Bull. **33**(3), 33–41 (2010)
5. FORCE-11: Data Citation Synthesis Group: Joint Declaration of Data Citation Principles. FORCE11, San Diego, CA, USA (2014)
6. Pan, S.J., Yang, Q., et al.: A survey on transfer learning. IEEE Transactions on knowledge and data engineering **22**(10), 1345–1359 (2010)
7. Silvello, G.: A Methodology for Citing Linked Open Data Subsets. D-Lib Magazine **21**(1/2) (2015). https://doi.org/10.1045/january2015-silvello
8. Silvello, G.: Learning to Cite Framework: How to Automatically Construct Citations for Hierarchical Data. Journal of the American Society for Information Science and Technology (JASIST) **68**(6), 1505–1524 (2017)

9. Silvello, G.: Theory and Practice of Data Citation. Journal of the American Society for Information Science and Technology (JASIST) **69**(1), 6–20 (2018)
10. Thorisson, G.A.: Accreditation and attribution in data sharing. Nature Biotechnology **27**, 984—985 (2009)
11. Wu, Y., Alawini, A., Davidson, S.B., Silvello, G.: Data Citation: Giving Credit Where Credit is Due. In: Proc. of the 2018 SIGMOD Conference. pp. 99–114. ACM Press, New York, USA (2018). https://doi.org/10.1145/3183713