

Classifying Document Types to Enhance Search and Recommendations in Digital Libraries

Aristotelis Charalampous and Petr Knoth^(✉)

CORE, Knowledge Media Institute, The Open University, Milton Keynes, UK
{aristotelis.charalampous, petr.knoth}@open.ac.uk

Abstract. In this paper, we address the problem of classifying documents available from the global network of (open access) repositories according to their type. We show that the metadata provided by repositories enabling us to distinguish research papers, thesis and slides are missing in over 60% of cases. While these metadata describing document types are useful in a variety of scenarios ranging from research analytics to improving search and recommender (SR) systems, this problem has not yet been sufficiently addressed in the context of the repositories infrastructure. We have developed a new approach for classifying document types using supervised machine learning based exclusively on text specific features. We achieve 0.96 F1-score using the random forest and Adaboost classifiers, which are the best performing models on our data. By analysing the SR system logs of the CORE [1] digital library aggregator, we show that users are an order of magnitude more likely to click on research papers and thesis than on slides. This suggests that using document types as a feature for ranking/filtering SR results in digital libraries has the potential to improve user experience.

Keywords: Document classification · Academic search · Recommender systems for research · Text mining · Metadata quality · Document aggregation

1 Introduction

Over the last 15 years, there has been a significant growth in the number of institutional and subject repositories storing research content. However, each repository on its own is of limited use, as the key value of repositories comes from being able to search, recommend and analyse content across this distributed network. While these repositories have been established to store primarily research papers, they contain, in fact, a variety of document types, including theses and slides. Services operating on the content from across this repository network should be able to distinguish between document types based on the supplied metadata.

However, metadata inconsistencies are making this very difficult. As we show later in the study, ~62% of documents in repositories do not have associated

metadata describing the document type. Moreover, when document type is specified, it is typically not done using an interoperable vocabulary.

Consequently, digital library aggregators like CORE [1], OpenAIRE [2] and BASE [3] face the challenge of offering seamless SR systems over poor quality metadata supplied by thousands of providers. We hypothesise that by understanding the document type, we can increase user engagement in these services, for example, by means of filtering or re-ranking SR systems results.

In this paper, we develop a novel and highly scalable system for automatic identification of research papers, slides and theses. By applying this identification system, we analyse the logs of CORE’ SR systems to see if we can find evidence of users preferring specific document type(s) over others.

The contributions of the paper are:

- Presenting a lightweight, supervised classification approach for detecting *Research*, *Slides* and *Thesis*, based on a small yet highly predictive set of features extracted from textual descriptors of (scientific) articles, reaching an F1-score of 96.2% with the random forest classifier.
- A publicly exposed and annotated dataset [4] of approximately 11.5 *k* of documents for the sake of comparison and reproducibility.
- Proposing a modified CTR metric, balanced QTCTR, to analyse historical SR systems’ logs to evaluate user engagement with the proposed content types in digital library systems, showing our users’ inclination towards research and theses over slides.

The rest of the paper is organised as follows. Firstly, we discuss related work, followed by the presentation of our current data state. Secondly, we outline our approach and present results of the classification approach and the analysis of current user engagement using our modified CTR metrics. Finally, we end with a discussion before concluding the paper.

2 Related Work

The library community holds traditionally metadata records as a key enabler for resource discovery. Systems, such as BASE and WorldCat¹, have been almost solely relying on metadata in their search services until today. But as such approach, as opposed to services indexing the content, cannot guarantee metadata validity, completeness and quality, nor can achieve acceptable recall [1], some have started to believe that aggregative digital libraries have failed due to the interoperability issues facing OAI-PMH data providers. In fact, [5] specifically argues that the fact that BASE and OpenAIRE do not (or cannot) distinguish between document types of the records they harvest makes them “not as effective as users might assume”.

While automatic document categorisation using structural and content features has been previously widely studied [6–8], little work has been done on the

¹ <https://www.worldcat.org/>.

issue of document type categorisation in the context of digital libraries until the recent study Caragea et al. [9]. They experimented with (1) *bag-of-words*, (2) document *URL tokens* and (3) document *structural features* to classify academic documents into several types. Their set of 43 manually engineered *structural features* have shown significant performance gain over conventional *bag-of-words* models in these highly diverse data collections.

Unlike previous work in standard approaches to text categorisation, summarised in [10], we use a subset of file and text specific characteristics, selectively gathered from [9]. The reduced dimensionality, as a result of the subset's minimal size, allows for scalable integration in ingestion pipelines of SR systems. In addition to the previous work, our study is to our knowledge the first to understand whether the integration of these document type classification systems can lead to more effective user engagement in SR systems.

3 Data - Current State

CORE is a global service that provides access to millions of (open access) research articles aggregated from thousands of OA repositories and journals at a full text level. CORE offers several services including a search engine, a recommendation system, an API for text-miners and developers as well as some analytical services. As of April 2017, CORE provides access to over 70 million metadata records and 6 million full texts aggregated from 2,461 data providers. From the available metadata descriptors, a directly available field to categorise records, at a certain extent, is the *dc:subjects* field. While mostly available, currently 92% of cases, only a small minority contain clear descriptions of the document type. More specifically, ~30.0% of records are marked as **article**, ~7.3% are marked as **thesis** and 0% as **slides**. This means that we do not have any type document type indication for ~62% of our data.

Table 1 lists the top re-occurring terms that are most indicative of the three document types we are interested in. This provides empirical evidence of the poor adoption of interoperable document type descriptors across data providers. Finally, from the ~6 million full text entries that CORE contains, 8.5 million

Table 1. Most popular terms found in the **dc:subjects** field with >1% occurrence

Term name	Term frequency
Article	0.1366
info:eu-repo/semantics/article	0.0866
Journal articles	0.0385
Thesis	0.0205
info:ulb-repo/semantics/openurl/article	0.0017
info:eu-repo/semantics/doctoralthesis	0.0106
info:eu-repo/semantics/bachelorthesis	0.0101

unique *dc:subjects* field terms are currently recorded (one record can contain multiple subjects fields).

4 Approach

While one approach to address the problem of poor or missing document type descriptors can be to create guidelines for data providers, we believe this approach is slow, unnecessarily complex and does not scale. Instead, we aim to develop an automated system that infers the document type from the full text.

The assumptions we make for this study follow several observations on the textual features of documents stored in CORE:

- **F1: Number of authors:** The more authors involved in a study, the more likely a document is a research paper as opposed to slides or thesis.
- **F2: Total words:** These were tokenised from the parsed text content using the *nltk*[11] package. Intuitively, the lengthier a document is, in terms of total written words and amount of pages, the more likely it is a thesis.
- **F3: Number of pages:** Research papers tend to have a fewer number of pages compared to theses and slides.
- **F4: Average words per page:** Calculated as $\frac{\#total\ words}{\#total\ pages}$. The fewer words written per page on average, the more likely the document type is *slides*.

We extract F2-F4 from their respective pdf files with pdfMiner [12]. F1 is extracted from the supplied metadata. We then apply one of the classifiers, described later in Sect. 5.2, to predict the document type given these features.

5 Experiments

5.1 Data Sample

Our first goal was to create a sufficiently large ground truth dataset. Data labelling took place with a rule-based method applied to the CORE dataset. More specifically, we used a set of regular expressions on the *dc:subjects* field and the document’s title as follows:

- Subjects fields for which entries include the keyword “thesis” or “dissertation” were labelled as *Thesis*.
- Subjects fields for which entries do **not** include the keyword “thesis” or “dissertation” and their title does **not** include the keyword “slides” or “presentation” were labelled as *Research*.
- Subject fields for which entries do **not** include the keyword “thesis” or “dissertation” and their title includes the keyword “slides” or “presentation” were labelled as *Slides*.

While this rule-based labelling process produced a sufficiently large number of samples for the *Research* and *Thesis* classes, it has not yielded a satisfactory sample size for the *Slides* class. To address this issue, we have mined pdfs and metadata from SlideShare² using their openly accessible API.

We wanted the total size of the sample to satisfy two criteria, a confidence level of 95% at a confidence interval of 1%. The equation to calculate the necessary size of the data sample is:

$$n = \frac{Z^2 \hat{p}(1 - \hat{p})}{c^2} \quad (1)$$

where, Z is the Z score, \hat{p} is the percentage probability of picking a sample and c is the desired confidence interval. Given a Z score of 1.96 for a 95% confidence level, a confidence interval of 0.01 and a sample proportion p of 0.5 (used as it is the most conservative and will give us the largest sample size calculation), this equation yields $\sim 9.6k$ samples.

We have gathered these $9.6k$ samples and additionally extended the dataset by 20% to form a validation set, resulting in $11.5k$ samples. To produce a sample with a representative balance of classes, we limited slides to take up to 10% of the final dataset, 55% for research and the remaining 35% for theses entries. We also ensured that all the pdfs in the data sample are parsable by pdfminer.

Finally, we addressed the issue of missing values for feature F1, which SlideShare did not provide in over 97% of cases, by applying multivariate imputations [13]. To improve our knowledge of the feature distributions prior to applying the imputations for the *Slides* class, we relied on extra data from Figshare³.

To visualise the dimensionality and data variance in the resulting dataset, we have produced two and three dimensional projections of our data, using techniques introduced by [14]. On small datasets ($< 100k$ data points) these do not require much tuning of hyper-parameters and, out of manual inspection from a limited range of hyper-parameters, we decided to use perplexity of 30 and a θ of 0.5. As Fig. 1 suggests, there is sufficient evidence of data sparsity.

5.2 Feature Analysis and Model Selection

We have experimented with: Random Forest (RF), Gaussian Naive Bayes (GNB), k Nearest Neighbours (k NN), Adaboost with Decision trees (Adaboost) and linear kernel Support Vector Machines (SVM).

We followed a standard 10-fold cross-validation approach to evaluate the models with an extra 20% of the data left aside for model validation. The class balance discussed was preserved in each fold evaluation by applying stratified splits on both test and validation sets, simulating a representative distribution of categories in the CORE dataset. All features used were compared against their normalised and log-scaled counterparts to check for any possible performance improvements. We have also optimised for a small range of hyper-parameters

² <https://www.slideshare.net/>.

³ <https://figshare.com/>.

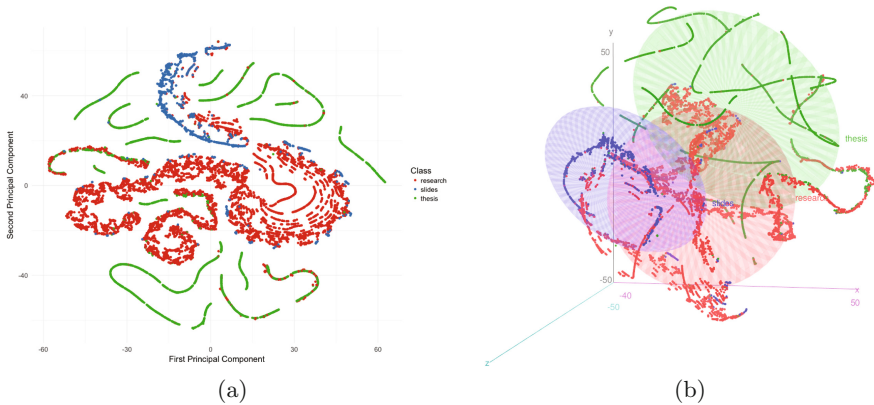


Fig. 1. Data variance visualisation using (a) two and (b) three dimensional projections on the corresponding principal components.

for each machine learning algorithm using parameter sweeps, recording the best achieved performance for each algorithm class. The evaluation results are presented in Table 3.

Two baseline models have been used to assess the improvement brought by the machine learning classifiers. The approaches used are:

- **Baseline 1:** Random class assignment with probability weights corresponding to the dataset’s class balance.
- **Baseline 2:** A rule-based approach based on statistically drawn thresholds for each feature and class respectively, using the upper 0.975 and lower 0.025 quantiles.

An analysis was carried out on the assembled dataset to form Baseline 2, based on feature distributions’ percentiles. Distributions from the sample dataset largely followed a right skewed normal distribution (Fig. 2), proving such a model should be a suitable candidate to evaluate against. To avoid overfitting, outliers were removed using Tukey’s method [15], which was preferred due to its independence on the data distribution, omitting values outside of the range:

$$(Q1 - 1.5 * IQR) > Y > (Q3 + 1.5 * IQR) \quad (2)$$

where, Y is the set of acceptable data points, $Q1$ is the lower quartile, $Q3$ is the upper quartile and $IQR = Q3 - Q1$ is the interquartile range.

The acquired thresholds for Baseline 2 are listed in Table 2. To assign a particular example a document type t , all its features must fall within the boundaries specified. When this method fails, we assign the majority class (*Research*).

Table 2. Percentile thresholds (upper 0.975 and lower 0.025 quantiles) for Baseline 2, following outlier removal.

Feature	Research	Document type	
		Slides	Thesis
F1	$1 \leq x \leq 5$	$1 \leq x \leq 8$	$==1$
F2	$1227 \leq x \leq 19, 151$	$94 \leq x \leq 7340$	$15, 184 \leq x \leq 210, 720$
F3	$3 \leq x \leq 41$	$1 \leq x \leq 75$	$47 \leq x \leq 478$
F4	$208 \leq x \leq 927$	$8 \leq x \leq 723$	$198 \leq x \leq 530$

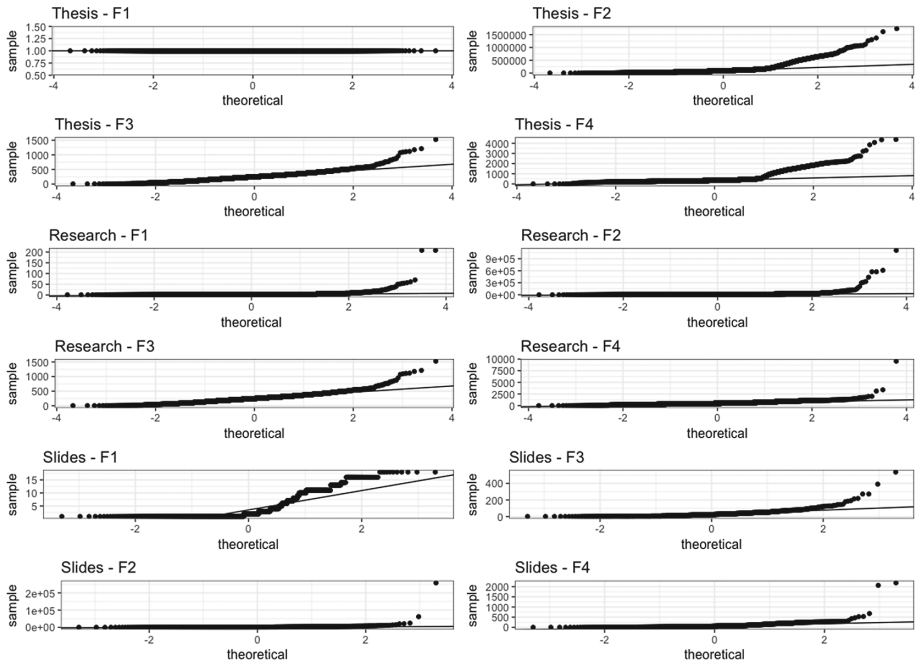


Fig. 2. Normal Q-Q Plots split by document type and feature.

5.3 Results

The evaluation results, presented in Table 3, show that all our models outperform the baselines by a large margin. However, baseline 2 demonstrates a perhaps surprisingly good performance on this task. Random forest and Adaboost are the top performers achieving about 0.96 in F1-score on both the test and validation sets. While we cannot distinguish which model is better at the 95% confidence level and 1% confidence interval, see Sect. 5.1, we decided to productionise random forest due to the model’s simplicity.

Table 3. Test and validation set results on weighted evaluation metrics across all algorithms.

	Measure	Algorithm						
		RF	GNB	kNN	Adaboost	SVM	Baseline 1	Baseline 2
Test results	Precision	0.962	0.9431	0.949	0.9580	0.8968	0.4926	0.5688
	Recall	0.9623	0.9414	0.9497	0.9569	0.8933	0.3270	0.4762
	F1-score	0.9623	0.9416	0.9496	0.9573	0.8695	0.3270	0.5154
Validation results	Precision	0.9567	0.9356	0.9453	0.9607	0.8435	0.5572	0.6362
	Recall	0.9553	0.9338	0.9454	0.9605	0.8741	0.4570	0.6565
	F1-score	0.9558	0.9337	0.9453	0.9606	0.8311	0.4570	0.5945

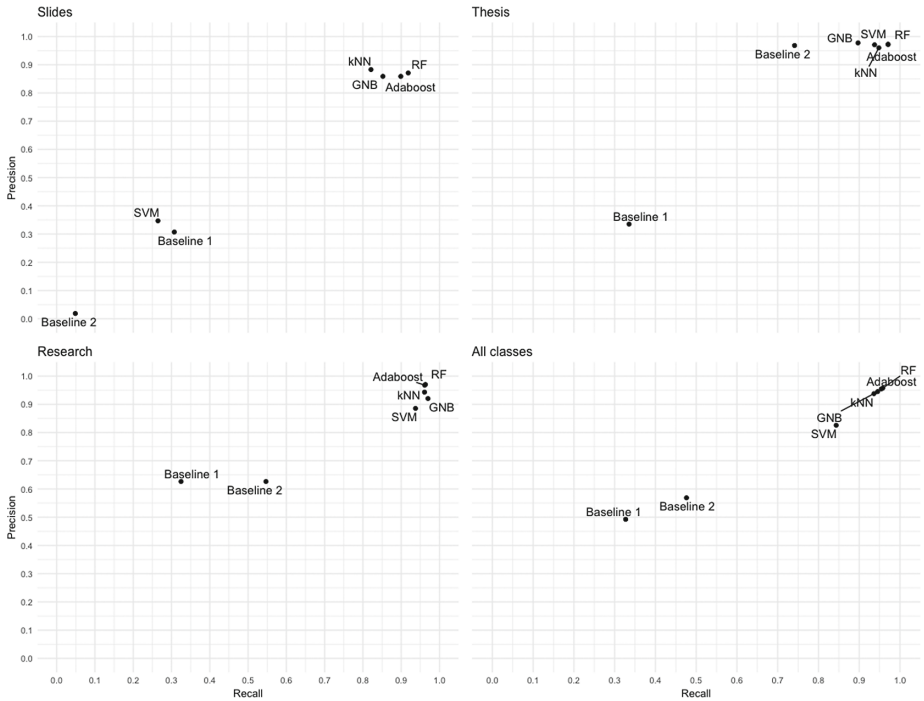


Fig. 3. Precision versus Recall for all algorithms on the test set split by class.

Figure 3 shows a breakdown of the final precision/recall performances according to the assigned document type. This indicates that a particularly significant improvement of the machine learning models over the baselines is achieved on the *Slides* class. However, as only about 10% of documents in the dataset are slides, the baselines are not so much penalised for these errors in the overall results.

To evaluate the importance of individual features, a *post-hoc* analysis was carried out. We fitted the models of our selected algorithms with a *single* feature

Table 4. Classifiers’ performance with individual feature groups across all algorithms on the test set in descending order, based on their contribution.

Features	Average weighted F1-score				
	RF	GNB	kNN	Adaboost	SVM
Only: F2	0.8825	0.7661	0.8702	0.8839	0.1868
Only: F3	0.8436	0.8412	0.8414	0.8424	0.8441
Only: F1	0.8007	0.6819	0.8007	0.8007	0.6441
Only: F4	0.7036	0.4745	0.6919	0.7018	0.3506
All features (RF)	0.9623	0.9416	0.9496	0.9573	0.8695

group at a time. In this scenario, we have recorded high precision performances. Individual feature contributions do not vary widely, except in the case of F4 and the overall performance of the SVM classifier. F1-3 are the most predictive features. We list our findings in Table 4.

6 Can the Model Help Improve User Engagement in SR Systems?

We applied the random forest model to classify existing content in CORE. Joining the document type information with CORE’s SR systems’ user logs, enabled us to analyse document type user preferences in CORE’s SR systems.⁴ We followed the intuition that if we can find that users prefer clicking in SR results on one document type over another, this will provide the argument for using document type information in SR systems to better serve the needs of these users.

A traditional metric to measure the popularity of a link is the Click-Through Rate (CTR), measured as:

$$CTR_T = \frac{|Clicks|}{|Impressions|} \quad (3)$$

However, we cannot use CTR directly to assess whether people are more likely to click on certain document types than others in the SR system results. This is because we serve, on average, 66.7% *Research*, 27.2% *Thesis* and 6.1% *Slides* impressions across our SR engines. Consequently, the CTR metric would be biased towards the *Slides* class. This is due to the fact that when an action is made on an impression set, the class most represented in the set will benefit from this action on average the least. Put differently, this is accounted to the class imbalance.

To address this problem, we extend CTR to put *impression equality* into perspective with the following process. We group impressed items in sets Q ,

⁴ It should be noted that as CORE provides thumbnails on its SR results pages, users get an idea of the document type prior to accessing it.

Table 5. Modified click-through rate metrics performance on CORE’s SR systems.

Metric	Engine	Impression set positions					
		Any position			Top position		
		Research	Slides	Thesis	Research	Slides	Thesis
QTCTR	Search	0.13685	0.01878	0.32358	0.03818	0.00389	0.01829
	Recommender	0.00675	0.00074	0.00361	0.00482	0.00046	0.00204
RQTCTR	Search	0.08186	0.00142	0.10061	0.02284	0.00029	0.00569
	Recommender	0.00488	0.00003	0.00079	0.00348	0.00002	0.00045

reflecting the documents served following a query submission (in case of the recommender, the query is a document with respect to which we recommend)⁵. We assign to each impression set a type q_t based on the types of document(s) clicked in the results list. In case multiple clicks to distinct document types are made in response to a query, we generate multiple impression sets derived from it, each assigned to one of them.

We then calculate the *Query Type Click-Through Rate (QTCTR)* as a fraction of the number of queries which resulted in a click to a given document type over the number of all queries:

$$QTCTR = \frac{|Q_T|}{|Q|} \quad (4)$$

QTCTR tells us the absolute proportion of queries that result in clicking on a particular document type. We can regularise/normalise *QTCTR* to reflect the imbalance of impression types, forming the *Regularised Query Type Click-Through Rate (RQTCTR)*. We include impression sets with no interaction in this calculation.

$$RQTCTR = \frac{|Q_T|}{|Q|} * \frac{|Impressions_T|}{|Impressions|} \quad (5)$$

The *QTCTR* and *RQTCTR* values from the CORE’s SR systems, for the three different document types, are presented in Table 5. The shows that there is noteworthy difference in preference for *Research* type documents and *Thesis* over *Slides* by an order of one magnitude. This is true for clicks generated on any document in an impression set and when the click was on top positioned document. The *QTCTR* results also reveal that many people in CORE are looking for theses. We believe this is due to the fact that CORE is one of the few systems (in not the only one) that aggregates theses from thousands of repositories at a full-text level.

⁵ The number of impressions generated in response to a query can vary across queries. In our case, it can be from zero to ten for search and from zero to five for the recommender.

7 Scalability Analysis

There exists a linear relationship between the number of features (N) and prediction latency [16], expressed with the complexity of $O(N * M)$, where M are the number of instances. The low number of features and model complexity, with our deployed model having < 10 trees and < 5 maximum nodes for each, the latency amounts to slightly over 0.0001 seconds per prediction⁶. Due to CORE's continuously ongoing repository harvesting processes, the minimal feature extraction requirements will allow for new additions to be streamlined immediately after their processing, in comparison with the latency associated with the feature extraction process expected from [9]. This indicates the high scalability of our approach and applicability across millions of documents.

8 Future Work

In promoting the current solution within CORE's systems, and making it accessible to users worldwide, we aim to:

- Expose document type classification models as a service, with online model updating, through CORE's public API.
- Boost *Research* documents in our SR engines and negatively boost *Slides* to aid faster retrieval of preferred content.
- Evaluate the shift of user engagement as a direct effect of such changes in our services and adjusting our search/recommendation strategies accordingly.
- Enhance user engagement analysis by cross-validation of our observations here metrics such as the *dwelt time*, a metric proven to be less unaffected by position, caption or other form of bias in SR results [17].
- Extend the model in further iterations to also discern between sub-types of the *Research* and *Slides* classes, such as theoretical, surveys, use case or seminal research papers as well as slides corresponding to conference papers and lecture/course slides respectively.

9 Conclusions

We have presented a new scalable method for detecting document types in digital libraries storing scholarly literature achieving 0.96 F1-score. We have integrated this classification system with the CORE digital library aggregator. This enabled us to analyse the SR system logs of to assess whether users prefer certain document types. Using a our Regularised Query Type Click-Through Rate (RQTCTR) metric, we have confirmed our hypothesis that the document type can contribute in finding a viable solution to improving user engagement.

Acknowledgements. This work has been partly funded by the EU OpenMinTeD project under the H2020-EINFRA-2014-2 call, Project ID: 654021. We would also like to acknowledge the support of Jisc for the CORE project.

⁶ This excludes network overhead from the API call and the feature extraction process.

References

1. Knoth, P., Zdráhal, Z.: CORE: three access levels to underpin open access. *D-Lib Mag.* **18**(11/12) (2012)
2. Rettberg, N., Schmidt, B.: Openaire-building a collaborative open access infrastructure for european researchers. *Liber Q.* **22**(3) (2012)
3. Summann, F.: Bielefeld academic search engine: a scientific search service for institutional repositories. In: *Open Scholarship 2006 Conference* (2006)
4. Classifying document types to enhance search and recommendations in digital libraries - Dataset, https://figshare.com/articles/Classifying_document_types_to_enhance_search_and_recommendations_in_digital_libraries/4834229. Accessed 21 Apr 2017
5. Poynder, R.: Q&A with CNI's Clifford Lynch: Time to re-think the institutional repository? *The Open Access Interviews* (2016)
6. Sebastiani, F.: Machine learning in automated text categorization. *ACM Comput. Surv. (CSUR)* **34**(1), 1–47 (2002)
7. Qi, X., Davison, B.D.: Web page classification: features and algorithms. *ACM Comput. Surv. (CSUR)* **41**(2), 12 (2009)
8. Ghosh, S., Mitra, P.: Combining content and structure similarity for xml document classification using composite SVM kernels. In: *19th International Conference on Pattern Recognition, ICPR 2008*, pp. 1–4. IEEE (2008)
9. Caragea, C., Wu, J., Gollapalli, S.D., Giles, C.L.: Document type classification in online digital libraries. In: *AAAI*, pp. 3997–4002 (2016)
10. Aphinyanaphongs, Y., Fu, L.D., Li, Z., Peskin, E.R., Efstathiadis, E., Aliferis, C.F., Statnikov, A.: A comprehensive empirical comparison of modern supervised classification and feature selection methods for text categorization. *J. Assoc. Inf. Sci. Technol.* **65**(10), 1964–1987 (2014)
11. Bird, S.: NLTK: the natural language toolkit. In: *Proceedings of the COLING/ACL on Interactive Presentation Sessions*, pp. 69–72. Association for Computational Linguistics (2006)
12. Shinyama, Y.: Pdfminer: Python PDF parser and analyzer (2015), <http://www.unixuser.org/euske/python/pdfminer/>. Accessed 08 Apr 2017
13. Buuren, S., Groothuis-Oudshoorn, K.: Mice: Multivariate imputation by chained equations in r. *J. Stat. Softw.* **45**(3) (2011)
14. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9**, 2579–2605 (2008)
15. Tukey, J.W.: Comparing individual means in the analysis of variance. *Biometrics*, 99–114 (1949)
16. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: 7. computational performance - scikit-learn 0.18.1 documentation, http://scikit-learn.org/stable/modules/computational_performance.html. Accessed 08 Apr 2017
17. Kim, Y., Hassan, A., White, R.W., Zitouni, I.: Modeling dwell time to predict click-level satisfaction. In: *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, pp. 193–202. ACM (2014)