BD2K ERuDIte: the Educational Resource Discovery Index for Data Science

José Luis Ambite
Information Sciences Institute
Univ. of Southern California
4676 Admiralty Way
Marina del Rey, CA, USA
ambite@isi.edu

Lily Fierro
Information Sciences Institute
Univ. of Southern California
4676 Admiralty Way
Marina del Rey, CA, USA
lfierro@isi.edu

Florian Geigl[*]
Graz University of Technology
Graz, Styria, Austria
florian.geigl@gmail.com

Jonathan Gordon
Information Sciences Institute
Univ. of Southern California
4676 Admiralty Way
Marina del Rey, CA, USA
jgordon@isi.edu

Gully APC Burns
Information Sciences Institute
Univ. of Southern California
4676 Admiralty Way
Marina del Rey, CA, USA
burns@isi.edu

Kristina Lerman
Information Sciences Institute
Univ. of Southern California
4676 Admiralty Way
Marina del Rey, CA, USA
lerman@isi.edu

ABSTRACT

The field of data science has developed over the years to enable the efficient integration and analysis of the increasingly large amounts of data being generated across many domains, ranging from social media, to sensor networks, to scientific experiments. Numerous subfields of biology and medicine, such as genetics, neuroimaging, and mobile health, are witnessing a data explosion that promises to revolutionize biomedical science by yielding novel insights and discoveries. To address the challenges posed by biomedical big data, the National Institutes of Health (NIH) launched the Big Data to Knowledge (BD2K) initiative (`datascience.nih.gov`). An important component of this effort is the training of biomedical researchers. To this end, the NIH has funded the BD2K Training Coordinating Center (TCC). A core activity of the BD2K TCC is to develop a web portal (`bigdatau.org`) to provide personalized training in data science to biomedical researchers.

In this paper, we describe our approach and initial efforts in constructing ERuDIte, the Educational Resource Discovery Index for Data Science, which powers the BD2K TCC web portal. ERuDIte harvests a wealth of resources available online for learning data science, both for beginners and experts, including massive open online courses (MOOCs), videos of tutorials and research talks presented at conferences, textbooks, blog posts, and standalone web pages. Though the potential volume of resources is exciting, these online learning materials are highly heterogeneous in quality, difficulty, format, and topic. As a result, this mix of content makes the field intimidating to enter and difficult to navigate. Moreover, data science is a rapidly evolving field, so there is a constant influx of new materials and concepts. ERuDIte leverages data science techniques to build the data science index. This paper describes how ERuDIte uses data extraction, data integration, machine learning, information retrieval, and natural language processing techniques to automatically collect, integrate, describe and organize existing online resources for learning data science.

Keywords

Information Integration; Machine Learning; Online Educational Resources

1. INTRODUCTION

The National Institutes of Health (NIH) launched the Big Data to Knowledge (BD2K) initiative (`datascience.nih.gov`) to fulfill the promise of biomedical "big data" [7]. NIH recognized that *"The ability to harvest the wealth of information contained in biomedical Big Data will advance our understanding of human health and disease; however, lack of appropriate tools, poor data accessibility, and insufficient training, are major impediments to rapid translational impact".*[1] The NIH BD2K program has funded 15 major centers[2] to investigate how data science can benefit diverse fields of biomedical research including genetics, neuroimaging, precision medicine, and mobile health. Ensuring that the advances produced by these centers, and other research efforts, permeate the biomedical research community and yield the expected benefits for human health, requires a significant increase in the number of biomedical researchers trained in data science. To address this need, the NIH has funded the BD2K Training Coordinating Center (TCC).

Data science demands knowledge from many branches of mathematics and computer science, notably statistics and

[*] Florian Geigl's work was performed as a visiting PhD student at the USC Information Sciences Institute.

WWW 2017 Companion, April 3–7, 2017, Perth, Australia.
ACM 978-1-4503-4914-7/17/04.
http://dx.doi.org/10.1145/3041021.3053060

[1] `https://datascience.nih.gov/bd2k`

[2] `https://datascience.nih.gov/bd2k/funded-programs/centers`

machine learning, and can be applied to multiple fields of study. Given the field's interdisciplinary nature and its growing popularity, many open learning resources have been published on the Web for anyone interested in learning about data science. However, these resources vary greatly in quality, topic coverage, difficulty and presentation formats, making entry into the world of data science confusing and daunting for learners.

To address these challenges, the BD2K Training Coordinating Center is developing a web portal (`bigdatau.org`) to provide a dynamic, personalized educational experience for biomedical researchers interested in learning about data science. The portal is powered by ERuDIte, the Educational Resource Discovery Index for Data Science, an enhanced collection of existing web-based training materials on data science. In order to build ERuDIte, we are developing novel, automated methods to identify, collect, integrate, describe, and organize web-based learning resources. In this paper, we describe several steps of this process.

In the collection stage, we have built a web-scraping framework that allows us to rapidly incorporate new sources and extract relevant data from them. In the integration stage, we have designed a unified schema for learning resources to integrate heterogeneous data into a single, consistent model. Under this model, the system also exposes the metadata of learning resources as linked data [4], so these resources can be easily cross-referenced by other web services. In the description stage, ERuDIte uses methods from machine learning, information retrieval, and natural language processing to tag resources with concepts from a hierarchical, multi-dimensional ontology designed to provide an extensible, lightweight description of core aspects of the field of data science.

In summary, both in its design and in its creation, ERuDIte uses the concepts and methods of the data science field that it aims to teach. ERuDIte will enable students and researchers to make the best use of the diverse data science learning resources available online.

2. BUILDING ERuDIte

Since ERuDIte is itself a data science project, its construction reflects some of the key stages in the data science workflow, namely data collection, integration, modeling, and visualization. We describe these processes next.

2.1 Resource Collection

Resource quality and relevance are essential to the development of ERuDIte. Consequently, our initial resource collection focused on curated, reliable sources. While some sources provide resource data through public APIs (e.g., `coursera.org`, `udacity.com`), most sources require scraping of websites intended for human navigation. For this, we built a modular framework using the popular Python packages BeautifulSoup and Dryscrape to handle both static websites and dynamic, JavaScript-based pages, which have historically been problematic.

In this framework, each source website is handled by a module designed for the site's structure and idiosyncrasies. These require some manual authoring, but, once created, the site-specific module automatically collects resource data. The scraping framework is packaged as a Docker image so it can be used without locally managing its dependencies. As a result, we were able to increase our resource collection efforts quickly because team members could simultaneously build new site-specific modules without disturbing the core infrastructure of the scraping framework.

To date, we have collected a total of 8,600 resources, which vary in granularity from individual videos to online courses that include multiple video lectures and associated training material. Table 1 describes the current sources, the number of learning resources per source, and the types of information extracted, such as resource descriptions, video transcripts, and supporting slides or other written materials.

2.2 Resource Integration

To integrate the heterogeneous resource data, we designed a single metadata standard to represent learning resources in the ERuDIte domain. To develop our standard, we reviewed and incorporated existing standards to facilitate cross-institution data sharing, including classes and properties from the Dublin Core,[3] Learning Resource Metadata Initiative (LRMI),[4] IEEE's Learning Object Metadata (LOM),[5] eXchanging Course Related Information (XCRI),[6] Metadata for Learning Opportunities (MLO),[7] and Schema.org vocabularies. Our standard has three classes: *LearningResource* (with 27 properties), *Person* (with 8 properties), and *Provider* (with 10 properties).

We implemented this model in two ways in the ERuDIte system. Internally, we store all the course metadata into a relational database. Externally, in `bigdatau.org`, each web page for an individual learning resource includes its metadata in the JSON-LD[8] format to facilitate data exchange and indexing by search engines.

2.2.1 Integrated Resource Database

Our relational database uses views to map source tables to our standard schema in order to remain flexible for any future changes in the schema. The scraping framework outputs source-specific tables, and the views in the database integrate the source data into a single schema model. We then use an additional reporting materialized view that joins relations defined by the schema to form a composite table that generates the data for resource detail pages for display and use on the BD2K TCC web portal (`bigdatau.org`). We generate an Elasticsearch (`elastic.co`) index from a query to this table, and that index powers the search interface on the web portal. The resources are also tagged with concepts from an ontology (cf. Section 2.3) that are used in a faceted search interface in the web portal.

2.2.2 Learning Resource Metadata as Linked Data

The Linked Data movement [4] seeks to make the data available on the web not only readable to humans but also to machines. The JSON-LD format is a popular way to insert structured data into regular web pages and contribute to the web of Linked Data. These structured data snippets can then be easily extracted by external tools and indexed by search

[3] `http://dublincore.org`

[4] `http://lrmi.dublincore.net`

[5] `https://standards.ieee.org/findstds/standard/1484.12.1-2002.html`

[6] `http://shop.bsigroup.com/ProductDetail/?pid=000000000030259242`

[7] `https://joinup.ec.europa.eu/catalogue/asset_release/metadata-learning-opportunities-mlo-advertising`

[8] `http://json-ld.org/`

Table 1: Currently Indexed Learning Resources

Provider/Source	Types	Total	With Descriptions	With Transcripts	With Slides or Documents
BD2K	Video, Written	251	219	13	15
edX	Course, Video	100	99	90	73
Coursera	Course, Video	84	84	64	62
Udacity	Course, Video	17	17	17	0
VideoLectures	Video	7796	5545	165	4376
YouTube	Video	69	55	0	0
ELIXIR	Course, Written	240	48	0	0
Bioconductor	Course, Written	5	2	0	0
Cornell Virtual Workshop	Course, Written	38	19	0	0
Total		**8600**	**6088**	**349**	**4526**

engines. In particular, Google encourages the use of JSON-LD over the schema.org vocabulary for this purpose.[9] In a spirit of open data sharing, we expose all the metadata for each of the learning resources in the ERuDIte collection as Linked Data in the JSON-LD format. As part of our integration pipeline, we developed an automated mapping functionality from the Resource Database directly to a JSON-LD format using our previous work on data exchange embodied in the Karma system [11]. Augmenting our published learning resources with JSON-LD structured data allows current and future collaborators to easily cross-reference any resource we collect, increasing data interchangeability across global efforts for educational resource indexing.

2.2.3 Global Schemas for Learning Resources

There are a variety of large-scale efforts across the world developing training resources, including MOOC providers as well as large research consortia like the BD2K program. One effort of particular importance in the biomedical space is the ELIXIR consortium,[10] which seeks to provide a distributed infrastructure for life-science across Europe, in a spirit akin to the NIH BD2K Initiative. The ELIXIR Programme includes a training component, the Training e-Support System (TeSS), analogous to the ERuDIte index in the BD2K TCC.

We have established a collaboration with ELIXIR TeSS to develop joint metadata standards for learning resources and to share data synergistically. Beyond this collaboration, we want to help inform global learning resource standards. Since Schema.org is one of the largest and most popular standards in use today, researchers from both projects have joined the World Wide Web Consortium (W3C) Schema Course Extension Group[11] in order to participate in the design of Schema.org's *Course* class extension, which we expect will provide the core metadata to describe learning resources.

2.3 Resource Modeling

As a first level of organization, we designed a hierarchical, multi-dimensional ontology to provide descriptions of the learning resources. This ontology provides learners with concepts that can assist them with resource exploration and discovery.

In the design of the ontology, we followed a multi-pronged approach. First, we identified a collection of concepts based on our knowledge of the data science domain and organized them hierarchically along six dimensions, which were defined and agreed upon by the authors based on the different facets that learners would want to use as filters on the resource collection. Second, we collected and reviewed the categories used to describe resources in each of the existing sources (e.g., `videolectures.org` provides a categorization of its video collection). Finally, we used two semi-automated methods to refine and extend the ontology.

As a first semi-automated method, we developed a system that analyzes the textual information associated with the learning resources (including titles, descriptions, syllabi, transcripts, slides, etc.) to automatically generate concepts from bigrams, trigrams, nouns, and shallow noun phrases[12] extracted from sentence trees constructed by the Stanford Parser [1]. In evaluating these automatically identified concepts, we found that shallow noun phrases from the parser provided the richest terms. We reviewed the 8,160 automatic concepts from the parser and eliminated ambiguous and irrelevant ones. We also added tags related to the depth, domain, and format of the course. This process identified a total of 861 tags.

As a second semi-automated method, we used non-negative matrix factorization [10] to discover topics in our resources. We analyzed the most significant words associated with each topic and then defined a concept for each of the topics. Much of this analysis confirmed the concepts identified earlier, but it also yielded ten additional tags.

To converge to our current ontology, we filtered these concepts using the following criteria:

1. Is there enough support for the concept within our resource collection? (Currently, we require more than five resources to be relevant to the concept.)
2. Does the proposed concept capture an abstracted phrase or concept that cannot be automatically extracted from text (i.e, that it would not be easily found by an information retrieval search over the resource text)?
3. How does the proposed tag impact a user's ability to discover a resource?
4. Does a clear definition for the concept exist?

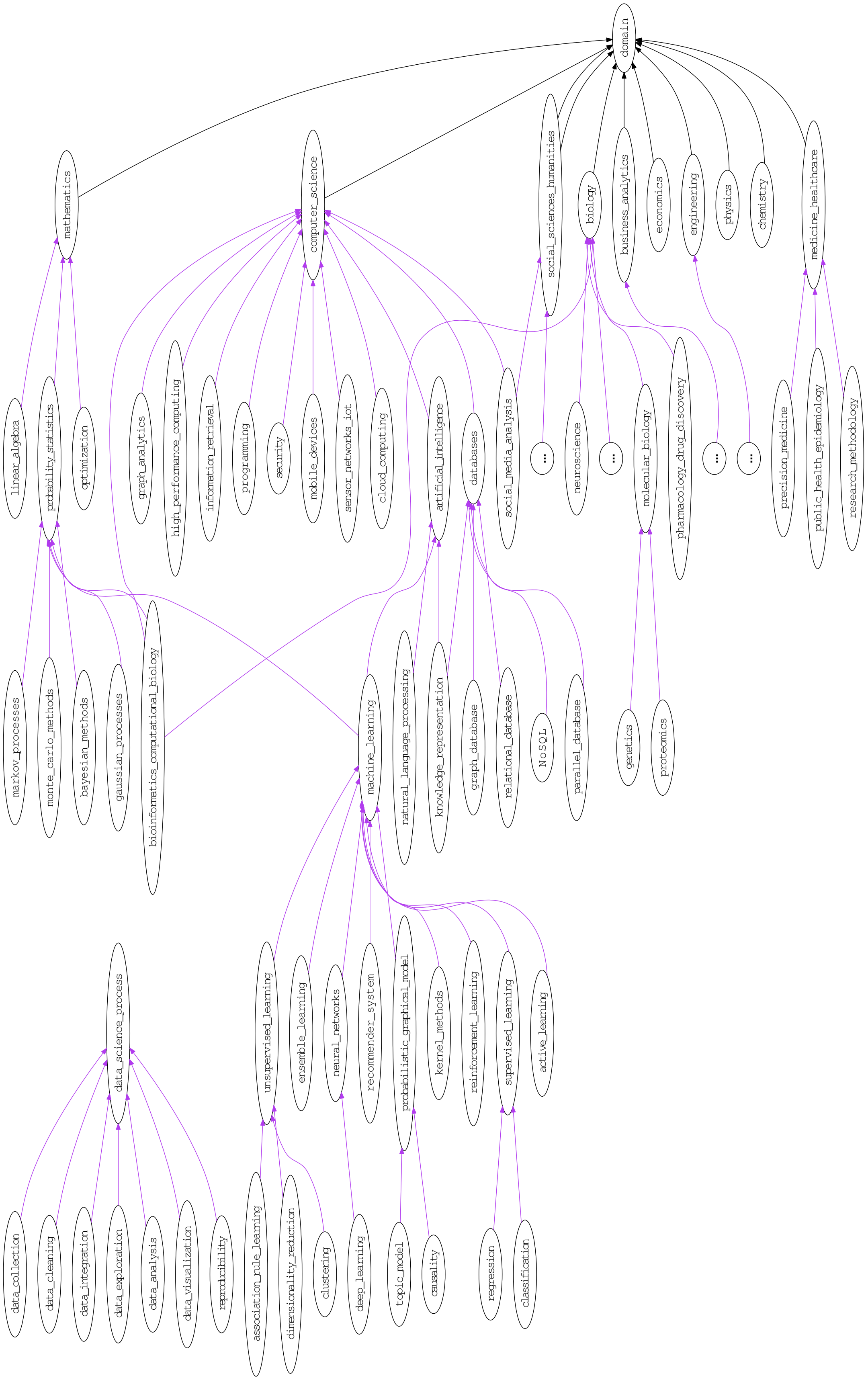
[9] https://developers.google.com/search/docs/guides/intro-structured-data

[10] https://www.elixir-europe.org

[11] https://www.w3.org/community/schema-course-extend

[12] We define "shallow noun phrases" as ones constructed with words at a single node level in the parse tree representation of resource descriptions.

Figure 1: All Data Science Process concepts and an excerpt of concepts from the Domain dimension. See `http://bigdatau.org/explore_erudite` and `http://bioportal.bioontology.org/ontologies/DSEO` for all concepts in the concept ontology.



5. Can the proposed concept be automatically predicted? (cf. Section 2.4).

This reduced the ontology to a total of 117 concepts, which we organized hierarchically along six dimensions. Figure 1 contains a selection of concepts from two dimensions, Domain and Data Science Process. A visualization of all concepts for the six dimensions is available at:
`http://bigdatau.org/explore_erudite`.

Each of the hierarchical dimensions of the concept ontology aims to answer a specific question a learner may have about a resource. These are listed below, along with how many of the 117 concepts currently identified fall under each dimension.[13]

Data Science Process (8) What stages of the data science process will this resource help me with?

Domain (74) What field of study does this resource focus on?

Datatype (18) What types of data are addressed in the resource?

Programming Tool (13) What programming tool is used in or taught by this resource?

Resource Format (2) How is this resource presented?

Resource Depth (2) How advanced is this resource?

We have represented the concept ontology for learning resources as a Simple Knowledge Organization System (SKOS)[14] vocabulary, with the hierarchal relationships encoded by the *broaderTransitive* property. We call this SKOS representation the Data Science Education Ontology (DSEO); it can be viewed and downloaded at:
`http://bioportal.bioontology.org/ontologies/DSEO`

2.4 Automatic Concept Assignment (Tagging)

In order to scale up ERuDIte, we need to develop automated methods to assign concepts from our ontology to the collected learning resources (i.e., tagging). For this purpose, we explored both machine learning and information retrieval methods. We defined an experimental procedure that used the same source data, cross-validation folds, and performance measurements for every method tested. We developed a gold standard by manually tagging 413 resources focused on data science, including massive open online courses (MOOCs) from Coursera, Udacity, and edX, and videos selected from Videolectures.net. For the inputs to the methods, we created text documents for each resource consisting of the resource title, subtitle, description, and syllabus, which were then vectorized as bag-of-words TF–IDF vectors.

Our performance metric is the F1 score [14], which is the harmonic mean of precision (positive predictive value) and recall (sensitivity). We calculate the average F1 metric over 5-fold cross-validation, and we perform a grid search over each method's specific hyperparameters with scikit-learn [8].[15] Specifically, for each tag, we calculated the F1 score of the validation fold. Then, we calculated the weighted average F1 across all tags with the weights equal to the number of true positives of each tag in the validation fold. Then, we computed the average of the weighted F1 average across all validation folds to get the final performance metric for between-method comparison. To predict if a concept applies to a resource, we trained the systems under two conditions:

1. *Exact Tag*: We train using the resources specifically tagged with a given concept.

2. *Tag & Descendants*: We exploit the hierarchical nature of our ontology by using as positive training examples the resources tagged with the exact concept and all the resources that are descendants of the concept in the hierarchy. For example, when predicting "machine learning", we include resources tagged with "machine learning" as well as those tagged with "neural networks", which is a descendant.

2.4.1 *Machine Learning Methods*

We used a one vs. rest approach to train random forest, multinomial naïve Bayes, logistic regression, support vector machines, and k-nearest neighbors classifiers. We show the training flow for the classifiers in Figure 2.

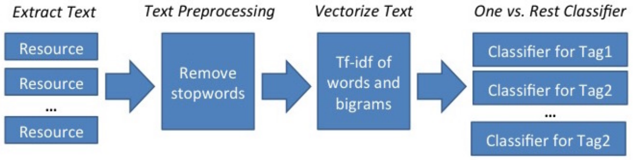


Figure 2: Processing workflow for training automated tagging using machine learning methods

Table 2 shows the performance for each classifier. For each of the classifiers, we did use non-negative matrix factorization (NMF) [10] to produce reduced document vectorizations, but we found that, across all classifiers, the TF–IDF representation performed best. Several classifiers performed comparably, but logistic regression produced the best results overall, with an F1-score of 0.73 when trained on exact tags, and of 0.83 when exploiting the concept hierarchy.

Table 2: Concept Assignment (Tagging): Machine Learning Methods (F1 Score)

Classifier	Exact Tag	Tag & Descendants
Random forest	0.66	0.75
Multinomial naïve Bayes	0.70	0.80
Logistic regression	**0.73**	**0.83**
Support vector machines	**0.73**	0.81
k-nearest neighbors	0.70	0.79

[13] The top-level concepts for the Data Science Process and Programming Tool dimensions can be assigned to resources, while the top-level names of the other dimensions are just used for organization.

[14] `https://www.w3.org/2004/02/skos`

[15] Tables 4 and 5 in the Appendix show the hyperparameter ranges used in the grid searches and the best values for the logistic regression method and the TF–IDF information retrieval method.

2.4.2 *Information Retrieval Methods*

For the tagging task, we also experimented with using ranked similarity between resource documents as a way to assign tags. In this approach, we first vectorized each resource document in the training set. Second, for each incoming resource in the validation fold, we compared the incoming resource vector to all resource vectors in the training set that belong to a tag. Third, we aggregated the similarities for all of the comparisons, and that aggregated value is the similarity metric used for that incoming resource-tag pair. Finally, once a similarity value has been calculated for every resource-tag pair, we sort the similarities and then use a rank cutoff to assign the top-n most similar tags to a resource. Figure 3 shows a diagram of this training and validation approach.

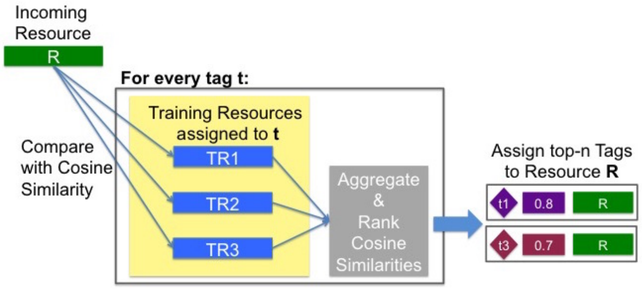

Incoming Resource
R
For every tag t:
Training Resources assigned to **t**
Compare with Cosine Similarity
TR1
TR2
TR3
Aggregate & Rank Cosine Similarities
Assign top-n Tags to Resource **R**
t1
0.8
R
t3
0.7
R

Figure 3: Processing workflow for training automated tagging with information retrieval methods

Table 3 shows the performance for the information retrieval methods for different methods of vectorizing the resource documents. We considered TF–IDF, as in the machine learning methods, plus two dimensionality reduction methods: NMF and latent semantic analysis (LSA) [2]. Unlike the machine learning classifiers, in the case of the information retrieval approaches, we determined that including the hierarchy in the training and validation was necessary in order to guarantee that enough vector comparisons can be made across all tags. Consequently, to compare the performance of the machine learning methods versus the information retrieval methods, we used the *Tag & Descendants* performance. Overall, with our current training data, the logistic regression classifier emerged as the best performing method for automated tagging.

Table 3: Concept Assignment (Tagging): Information Retrieval Methods (F1 Score)

Vectorization	Tag & Descendants
TF–IDF	**0.78**
NMF	0.77
LSA	0.77

2.5 Resource Visualization

Concept tags add an organizational structure for browsing resources on the BD2K TCC web portal, but we also want users to be able to explore resources through a visualization that conveys the landscape of resources in ERuDIte. For this, we used t-distributed stochastic neighbor embedding (t-SNE) [13] to reduce the vectorized representations into a two-dimensional space.

When determining the terms to use in the tagging ontology, we first used NMF topics to look for any underlying structure. In creating the t-SNE visualizations, we wanted to see if the NMF topics would facilitate some level of visual clustering, which would then allow learners to explore resources that were similar to each other. We experimented with NMF topic numbers and perplexity values (based on the observations of [15]), and we chose to only include resources that had both title and description in order to ensure that enough text was available to produce a reasonable vector to represent the resource's content. This created a corpus that contained 6,088 resources. To construct the NMF vectors, we created each resource document by concatenating the title, description, and syllabus text fields, applied TF–IDF onto the full corpus, and then applied NMF onto the resource vectorizations.

Figure 4 shows the t-SNE visualization with the clearest and semantically meaningful cluster structures (produced with 75 NMF topics and perplexity value of 50). We explored topic sizes of 25, 50, 75, and 100. Interestingly, the number of topics of the best visualization is close to the number of tags in the Domain dimension of our tagging ontology, suggesting that our tag selection matches the structure existing in our resources.

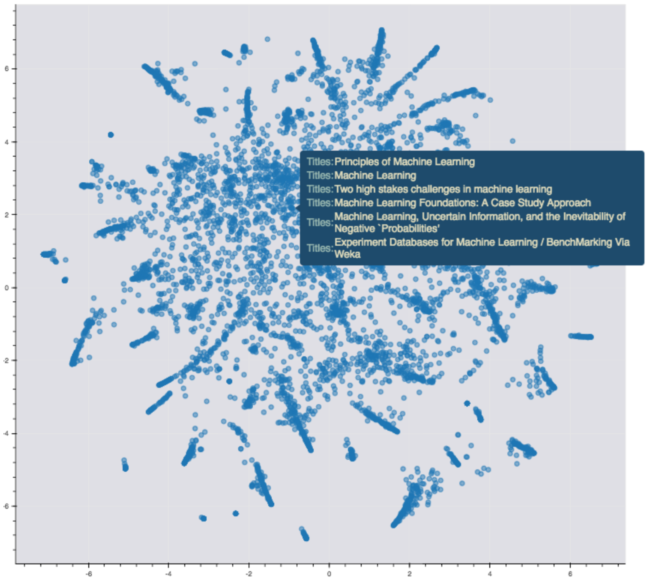


Figure 4: Visualizing the structure of Learning Resources, represented by NMF topic vectors, using t-SNE

3. ONGOING WORK

The ERuDIte system is under active development. To reach our vision for ERuDIte as a dynamically updated, personalized system suited for self-directed learning, we are pursuing the following research directions.

3.1 Automated Resource Identification

ERuDIte currently contains 8,600 resources, but our collection efforts have been skewed towards materials from manually-selected high-quality sources, such as MOOC providers. Much pedagogically valuable written material is available online. We plan to increase the number of relevant written documents indexed in ERuDIte. Similarly, YouTube provides many data science videos. We are expanding our resource collection efforts on written materials and YouTube videos and developing automated techniques for scoring the quality of resources based on resource metadata (views, likes, length, etc.) and instructor and provider publications and affiliations.

3.2 Curation and Continuous Improvement

At the BD2K TCC, we have a strong focus on automated techniques. However, since we want to ensure that we serve high-quality learning resources, we plan to introduce a level of human curation into our pipeline. We have begun the development of a curation interface that will reduce the effort required to tag resources. The curation interface will allow users to assess the quality of a resource and validate the concepts predicted by our algorithms, as well as to suggest missing concepts from our concept ontology. The curated concept/tags will be used to retrain our tagging algorithms and improve their performance. We plan to measure the improvements in tagging accuracy and on the efficiency of human curators.

While curation will initially be internal to the project, we envision later opening it to users of the web portal or crowdsourced workers, allowing us to re-train and validate our automated tagging algorithms at scale.

3.3 Dependencies and Prerequisites

In the current BD2K TCC web portal (`bigdatau.org`), learners can search through the resource index with keywords, can filter the resources based on our multi-dimensional, hierarchical concept ontology, and can obtain a recommendation of similar resources. However, we want to provide a stronger organizational structure that indicates which resources a learner should start with given a learning goal and state of expertise, and which resources should be studied before others. Consequently, we have begun to experiment with methods to extract resource dependencies and prerequisites. By conveying resource dependencies and prerequisites, we aim to provide a sequence of concepts and resources that will guide learners toward known learning paths as they develop and complete their own self-directed curriculum on the BD2K TCC web portal.

Members of the ERuDIte team previously developed TechKnAcq, a system that uses cross entropy to infer conceptual dependencies from collections of technical or scientific documents [3]. We have applied TechKnAcq methods on ERuDIte resources, but the initial results were not satisfactory. A likely explanation is that most of our learning resources offer less – and noisier – text than the journal articles and other publications organized by TechKnAcq. We plan to extract and automatically clean additional textual data from ERuDIte resources, including transcripts of videos and text found in associated materials, including lecture slides.

We are also exploring other approaches for learning resource dependencies, such as exploiting the ordered entries in course syllabi and the tables of contents in textbooks. The discovery of dependencies or prerequisites for general concepts or for individual resources is essential to creating a rich learning experience.

3.4 Personalization

We plan to explore personalization methods in ERuDIte through recommendations tailored for an individual user via collaborative filtering. To do this, we are instrumenting the web portal to collect user activity data. This will allow us to benefit from a large, consistently engaged user base to build our recommendation engine.

4. RELATED WORK

We briefly review work related to ERuDIte. There are a number of commercial "MOOC aggregators" (such as CourseBuffet, CourseTalk, TubeCourse, etc.), developed as social web applications, but the techniques for automatic identification, description, and organization of learning resources we propose in ERuDIte go beyond what these sites provide. The TechKnAcq project serves as an example of the possibility of such methods, attempting to structure the underlying organization of a pedagogical resource based on analyses of the content of that resource [3]. The concept hierarchies we use to describe resources can also be learned from existing resources [9]. For our visualization approach, we build on our previous work on the NIHMaps project [12], which provided a navigable map of all grants issued by NIH allowing users to explore the high-level structure of funded grants across several years. Other efforts have also used NMF to drive the creation of visual clusters. The multi-view NMF of [5] shows the potential to use more resource metadata in the generation of future resource visualizations. In the BD2K program there is a parallel effort, bioCADDIE, to catalog scientific datasets [6], but it is not focused on learning resources. ELIXIR's Training e-Support System (TeSS),[16] developed by ELIXIR-UK, has similar goals as the BD2K TCC. We are coordinating with ELIXIR TeSS to share resources and exploit synergies.

5. CONCLUSIONS

When looking at ERuDIte as its own data science project, we have made significant progress on the data collection and data integration steps, and we have begun the data exploration and data analysis steps. In the development of ERuDIte, so far, we have designed and implemented a flexible scraping framework, a unified schema, a tagging ontology, a visualization approach for resource exploration, and a collection of automated tagging algorithms. We are halfway toward completing the vision of making ERuDIte a platform that aggregates and organizes relevant resources and provides a personalized and engaging experience for the self-directed data science learner.

Although ERuDIte currently focuses on knowledge about data science, we expect that the ERuDIte platform can be applied to other fields. Most careers demand continuous, self-directed learning well outside of degree programs, and few tools exist to help learners navigate through the heterogeneous resources on the Web. Consequently, ERuDIte has the potential to expand interaction with an important subset of scholarly data: educational resources. Historically, when thinking about the web of scholars, we look at journal publications and citations, but now, in the age of digital learning, scholars also produce open-access educational resources, creating a source of data that connects, informs,

[16] `https://tess.elixir-europe.org/`

and educates not only scholars but also anyone interested in learning more about a field, concept, or technique. With this type of educational, scholarly data, the web of scholars can strengthen across disciplines, for understanding of others' work is easier through an open educational resource in comparison to a journal article, and can grow because many more people have access to the materials they need to learn in order to become scholars themselves.

6. ACKNOWLEDGMENTS

This work is supported by the National Institutes of Health under Grant 1U24ES026465-01[17]. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the National Institutes of Health.

We thank the rest of the BD2K TCC team: Jeana Kamdar, Sumiko Abe, Xiaoyu Lei, Avnish Bhattrai, Crystal Stewart, Caroline O'Driscoll, Carmen Tan, Shobhit Agarwal, and Krithika Ramaswamy for their support and their work on the TCC web portal. We thank Rochelle Tractenberg for her guidance in the process of defining skills and knowledge areas essential for data science. We thank Priyambada Jain for her contributions to website modules, and Pedro Szekely, Dipsy Kapoor and Amandeep Singh for support with the Karma system.

7. ADDITIONAL AUTHORS

Additional authors: John D. Van Horn (Stevens Neuroimaging and Informatics Institute, Univ. of Southern California, 2025 Zonal Ave, Los Angeles, CA, USA, email: `jvanhorn@usc.edu`).

8. REFERENCES

[1] D. Chen and C. D. Manning. A fast and accurate dependency parser using neural networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[2] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science (1986-1998)*, 41(6):391, 09 1990.

[3] J. Gordon, L. Zhu, A. Galstyan, P. Natarajan, and G. Burns. Modeling concept dependencies in a scientific corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 866–75. Association for Computational Linguistics, Aug. 2016.

[4] T. Heath and C. Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Synthesis Lectures on the Semantic Web. Morgan & Claypool Publishers, 2011.

[5] Y. Liu, Z. Huang, Y. Yan, and Y. Chen. Science navigation map: An interactive data mining tool for literature analysis. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, pages 591–6, New York, NY, USA, 2015. ACM.

[6] P. McQuilton, A. Gonzalez-Beltran, P. Rocca-Serra, M. Thurston, A. Lister, E. Maguire, and S.-A. Sansone. BioSharing: curated and crowd-sourced metadata standards, databases and data policies in the life sciences. *Database: the journal of biological databases and curation*, 2016.

[7] L. Ohno-Machado. NIH's big data to knowledge initiative and the advancement of biomedical informatics. *Journal of the American Medical Informatics Association (JAMIA)*, 193, 2014. doi: 10.1136/amiajnl-2014-002666.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[9] A. Plangprasopchok, K. Lerman, and L. Getoor. A probabilistic approach for learning folksonomies from structured data. In *Proceedings of the 4th ACM Web Search and Data Mining Conference (WSDM)*, Feb. 2011.

[10] F. Shahnaz, M. W. Berry, V. Pauca, and R. J. Plemmons. Document clustering using nonnegative matrix factorization. *Information Processing & Management*, 42(2):373–386, 2006.

[11] M. Taheriyan, C. A. Knoblock, P. Szekely, and J. L. Ambite. Semi-automatically modeling web APIs to create linked APIs. In *Proceedings of the ESWC 2012 Workshop on Linked APIs*, 2012.

[12] E. M. Talley, D. Newman, D. Mimno, B. W. Herr, H. M. Wallach, G. A. P. C. Burns, A. G. M. Leenders, and A. McCallum. Database of NIH grants using machine-learned categories and graphical clustering. *Nat. Meth.*, 8(6):443–4, June 2011.

[13] L. van der Maaten and G. Hinton. Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, 9:2579–605, Nov. 2008.

[14] C. Van Rijsbergen. Foundation of evaluation. *Journal of Documentation*, 30(4):365–373, 1974.

[15] M. Wattenberg, F. Viégas, and I. Johnson. How to use t-SNE effectively. *Distill*, 2016. `http://distill.pub/2016/misread-tsne`.

[17] http://dx.doi.org/10.13039/100000002

APPENDIX

A. GRID SEARCH HYPERPARAMETERS FOR AUTOMATED TAGGING

Table 4: Hyperparameters and Value Ranges for Logistic Regression Classifiers

Parameter	Parameter Description	Range	Best
Penalty	Normalization to use in penalization	[l1, l2]	l2
C	Inverse regularization strength	0.1–30	21
Intercept Scaling	When including intercept, the scaling of the intercept term	0.1–1.0	0.5
Class Weight	*Uniform* counts all classes of labels equivalently; *balanced* adjusts classes based on their frequencies	[uniform, balanced]	balanced
Prob. Threshold	Probability threshold to determine that a tag is assigned to a resource	0.1–0.9	0.4
Title Weight	Include title *title weight* times in resource document	1	1
Subtitle Weight	Include subtitle *subtitle weight* times in resource document	[0, 1]	0
Description Weight	Include description *description weight* times in resource document	1	1
Syllabus Weight	Include syllabus *syllabus weight* in resource document	[0, 1]	1
Stop Words	Stop word collection removal	[none, English]	English
Max. Document Frequency Threshold	Remove terms that occur in more than this proportion of resource documents	0.4–1.0	0.7
Min. Document Frequency Threshold	Remove terms that occur in less than this number of resource documents	3–10	8
N-gram Range	Include n-grams in vectorizations	$(1, 1)$ $(1, 2)$ $(1, 3)$	$(1, 2)$
NMF	Reduce vectorization with non-negative matrix factorization	[true, false]	false

Table 5: Hyperparameters and Value Ranges for TF–IDF Vector Comparison

Parameter	Parameter Description	Range	Best
Title Weight	Include title *title weight* times in resource document	2–4	4
Subtitle Weight	Include subtitle *subtitle weight* times in resource document	[1, 2]	1
Description Weight	Include description *description weight* times in resource document	[1, 2]	1
Syllabus Weight	Include syllabus *syllabus weight* in resource document	1	1
Stop Words	Stop word collection removal	[English]	English
Max. Document Frequency Threshold	Remove terms that occur in more than this proportion of resource documents	0.6–0.9	0.6
Min. Document Frequency Threshold	Remove terms that occur in less than this number of resource documents	2–5	2
N-gram Range	Include n-grams in vectorizations	$(1, 2)$ $(1, 3)$ $(1, 4)$	$(1, 2)$
Sublinear Term Frequency	Make term frequency equal to $1 + \log(\text{tf})$	[true, false]	true
Normalization	l1 is the Manhattan Distance, l2 is the Euclidean norm	[l1, l2]	l2
Inverse Document Frequency	Multiply term frequency by inverse document frequency	[true, false]	false
Similarity Aggregation	Use this aggregation function to assign resource–tag pairs	[mean, max, median, min]	max
Rank	Select up to *rank* value to assign tags to a resource	5–10	10