COCO: Semantic-Enriched Collection of Online Courses at Scale with Experimental Use Cases

Danilo Dessì, Gianni Fenu, Mirko Marras(✉), and Diego Reforgiato Recupero

Department of Mathematics and Computer Science,
University of Cagliari, V. Ospedale 72, 09124 Cagliari, Italy
{danilo_dessi,fenu,mirko.marras,diego.reforgiato}@unica.it

Abstract. With the proliferation in number and scale of online courses, several challenges have emerged in supporting stakeholders during their delivery and fruition. Machine Learning and Semantic Analysis can add value to the underlying online environments in order to overcome a subset of such challenges (e.g. classification, retrieval, and recommendation). However, conducting reproducible experiments in these applications is still an open problem due to the lack of available datasets in Technology-Enhanced Learning (TEL), mostly small and local. In this paper, we propose COCO, a novel semantic-enriched collection including over 43 K online courses at scale, 16 K instructors and 2,5 M learners who provided 4,5 M ratings and 1,2 M comments in total. This outruns existing TEL datasets in terms of scale, completeness, and comprehensiveness. Besides describing the collection procedure and the dataset structure, we depict and analyze two potential use cases as meaningful examples of the large variety of multi-disciplinary studies made possible by having COCO.

Keywords: Dataset · Online courses · Classification
Recommendation

1 Introduction

With the ever-increasing number of advanced personalized platforms and services available to support education, emerging technologies are reshaping how people learn in their everyday life, leading the global market of off-the-shelf education to growth and evolve towards online learning at scale [1]. More and more individuals and teams are leveraging it to continuously cultivate new skill sets and achieve personal or collective goals throughout their careers, while leading providers are offering large-scale on-demand access to their collections of online courses with varied contents, structures, requirements, objectives, instructors and prices [2].

The recent proliferation in number and scale of online courses has posed new challenges involving all the stakeholders. For instance, existing providers would automatically organize their online courses according to meaningful taxonomies

Á. Rocha et al. (Eds.): WorldCIST'18 2018, AISC 746, pp. 1386–1396, 2018.
https://doi.org/10.1007/978-3-319-77712-2_133

which facilitate retrieval, while instructors would find emerging teaching topics and develop courses on appealing subjects on the basis of the latest trends. Even more, learners would be driven along the overwhelming alternative courses. The solutions needed to overcome such challenges in online course delivering at scale depend on advanced infrastructures for storage, computation, and user interface, and computer scientists are drawn as a powerful medium to develop them. The literature has extensively addressed clustering [3], classification [4], retrieval [5] and recommendation [6] regarding individual educational resources (e.g. slides, videos, documents). On the other hand, the application of such techniques with the entire courses as targeted entities has been driven just by local institutions to support freshmen [7]. Only recently, the development under large-scale online environments has greatly progressed [8,9].

Despite the initial outcomes, different evolving problems remain to be solved. For instance, machine learning techniques need to combine both descriptive and content-based course features which require high-level semantic understanding. Even more, online courses at scale come with various languages, so algorithms are supposed to master cross-language capabilities. Similarly, recommendations targeted to learners should match their desired content with their requirements, goals, pedagogical, economic and temporal constraints. Moreover, reciprocal recommendations between learners and instructors are becoming appealing. One major obstacle in these directions is the lack of suitable datasets. To build a high-level semantic understanding, we need fine-grained information about courses. To test cross-language capabilities, we require courses provided in different languages. To provide meaningful recommendations, we need stakeholder interactions within courses. However, no dataset currently meets such conditions.

In this paper, we introduce COCO, a novel semantic-enriched Collection of Online COurses composed by more than 43 K courses distributed in 35 different languages, involving over 16 K instructors and 2,5 M learners who provided about 4,5 M ratings and 1,2 M comments. Furthermore, we provide two potential use cases where COCO can be handy, highlighting the issues which need to be faced.

The paper is organized as follows. First, Sect. 2 describes the collection procedure and the dataset structure together with some statistics. Then, Sect. 3 analyzes two use cases made possible by having COCO and demonstrates how they are promising and challenging. Section 4 compares COCO with the other existing datasets. Finally, Sect. 5 depicts some conclusions and future work.

2 The Proposed Dataset

COCO[1] is a research-purpose-only dataset which aims to support analysis, discussion, and design of tools and services in online learning. The dataset includes information collected from Udemy[2], one of the leading global marketplaces for online learning and teaching at scale. Unlike academic-oriented platforms driven to traditional coursework, Udemy enables experts in various areas to offer courses

[1] Please contact the authors by e-mail to obtain a copy of the dataset.

[2] https://www.udemy.com/.

at no charge or for tuition fees. In comparison with other online course platforms, no third-party control on reliability, validity, accuracy or truthfulness of the course content is performed. Collected data are verified, cleaned, and analyzed. All copyright and registered trademarks remain property of their owners.

2.1 Collection Methodology

The Udemy APIs[3] expose functionalities to help developers accessing content and building external applications. However, they are instructed to list only a subset of the over 40 K courses Udemy includes. Consequently, we developed a Selenium[4] crawler in Python to access the full Udemy catalog and build a complete and comprehensive dataset. We dumped it in November 2017.

The crawler is instructed to access the course catalog sublists associated to each category of the Udemy taxonomy, so that we first extract the association between each course and the corresponding categories while getting the link to the course description page. Each course has one first-level category and one second-level category. Each second-level category belongs to only one first-level category. Unlike traditional academic taxonomies, the courses are mapped by Udemy in daily-life-oriented categories (e.g. *Lifestyle*, *Language*, *Test Preparation*). Furthermore, each course is also described with a set of fine-grained tags. We extract the association between courses and tags using the same methodology previously employed to extract categories. However, the same course can appear in the course catalog sublist of more than one tag in that case.

Then, the crawler goes inside the description page of each course. To get an idea, an example course description page is made available here[5]. Each course description page presents the course identifier, the heading with the title and the short description of the course, aggregated statistics about received ratings and enrolled students, and the language in which the course is delivered. Udemy provides courses in more than 30 different languages. Then, different HTML boxes contain a bullet-list of course objectives (e.g. *build powerful fast user-friendly web apps*, *apply for high-paid jobs or work as freelancer*), a bullet-list of both pedagogical and technical course requirements (e.g. *Javascript and HTML fundamentals, a laptop with at least 6 GB RAM*), a long course description of around 500 words, and a bullet-list of possible target users (e.g. *students who want to learn how to build reactive web apps*) written by the instructors. The course description pages also include the list of lessons and their organization in chapters. Each lesson has a title, a 30/50-word description, and a format (e.g. *video* or *document*). Only a subset of lessons is freely available as preview. We collected their resource URL and, eventually, the URL of the video transcript. Furthermore, the course description pages list one or more instructors together with their id, job title and short biography. On the left-side, a HTML box depicts the current price. The crawler digests all such information.

[3] https://www.udemy.com/developers/.

[4] http://www.seleniumhq.org/.

[5] https://www.udemy.com/spark-and-python-for-big-data-with-pyspark/.

To extract the learners reviews, the crawler uses the Udemy API method aimed to return course reviews given the course identifier. Each review includes the learner id and the course id together with the timestamp, the rating ranging between 0 and 5 with step 0.5 and, optionally, a textual comment. It is worth to notice that learners give their comments in their own language, but no language label is provided to keep it. The mentioned API method does not release information regarding the instructor replies to learners reviews, so the crawler digests a copy of the same course reviews from the list presented at the bottom of the course description page. Differently from the mentioned API method, the course description page does not depict the review timestamps, but shows the instructor replies to such reviews. Then, the two copies of the same course reviews are merged. Moreover, the course description page depicts the full name of the learner who has made a given review. The crawler uses it on the fly to build the URL of the public profile of the learner and access it. Each public profile shows the courses where learners are enrolled and the wish-list in the case they have given consent to publicly share them. Finally, we label all the human-made textual attributes with their own language using Lang Detect[6], a free reliable language detector.

2.2 Semantic Enrichment Methodology

Course attributes and interactions with them embrace a wide range of human-made-based texts such as comments, requirements, objectives, descriptions, and video transcripts. Manipulating them in machine learning methods tailored for online courses requires high-level semantic understanding, going beyond traditional item-frequency features. To facilitate future experiments and comparisons, we first enriched such attributes with the Term Frequency Inverse Document Frequency (TF-IDF) features extracted by Scikit-Learn[7] v0.19. TF-IDF is one of the most popular term-weighting schemes that measures how important a word is to a document in a collection. Then, to stimulate research in high-level semantic understanding algorithms, we collected the features extracted by state-of-the-art cognitive tools. More precisely, we enriched the textual attributes with the following additional feature sets.

- Part-of-Speech (PoS) tags computed by the Natural Language Processing (NLP) tools of the Natural Language Toolkit[8] (NLTK), a leading platform for building programs working with human language.
- Keywords and concepts computed by the state-of-the-art Cognitive Computing (CC) tools included into the IBM Watson Natural Language Understanding APIs[9]. Each keyword is a set of one or more words relevant in the text, while each concept captures cross-domain content not explicitly cited.

[6] https://pypi.python.org/pypi/langdetect?.

[7] http://scikit-learn.org.

[8] http://www.nltk.org/.

[9] https://www.ibm.com/watson/services/natural-language-understanding/.

Traditional term-frequency-based methods like TF-IDF are easily computable, but they ignore roles that words play in sentences and semantic relations among them. By contrast, text representation with semantics accurately describes the text meaning, but requires higher computational cost due to the more complex underlying algorithms. To extract them, we employed tools intensively used into the existing state-of-the-art semantic approaches. The features are provided as they are, without applying feature selection or transformation methods.

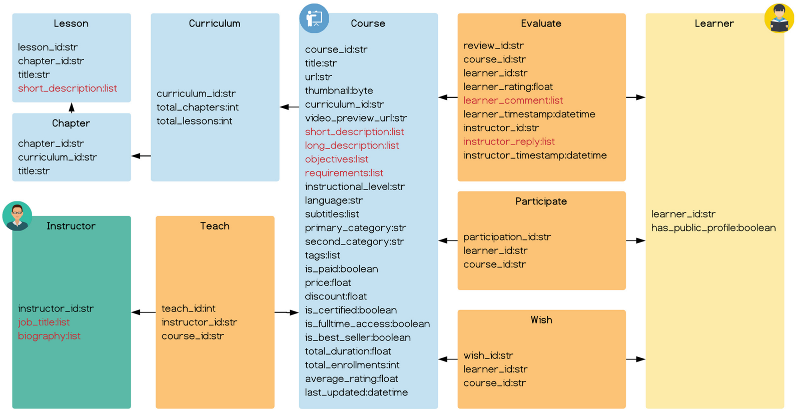


Fig. 1. *COCO Structure.* Boxes in green, blue and yellow show primitive entities. Orange boxes depict associations. The attributes in red are enriched with semantics.

2.3 Structure

COCO is a JSON-based collection whose structure in terms of entities and associations is depicted in Fig. 1. Text attributes have Unicode coding, while languages and timestamps hold ISO639-1 and ISO8601 standards, respectively.

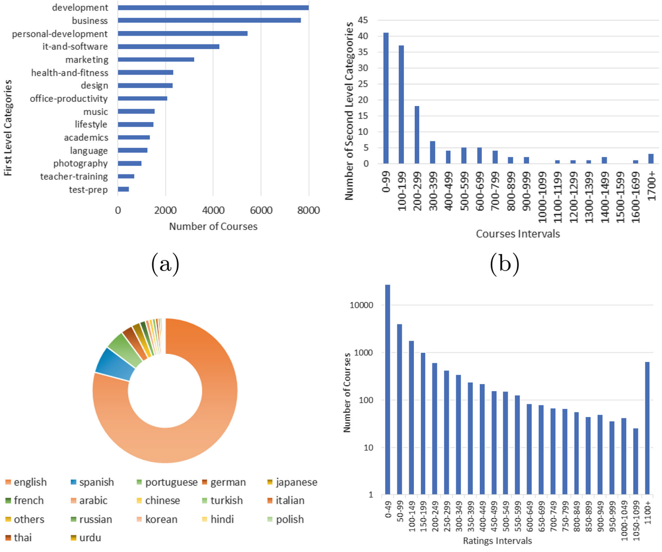
In COCO, *Course* is the most informative entity. First, *id* and *course URL* provide unique identification attributes. Then, the course is described by *short* and *long descriptions*. *Requirements* and *objectives* list technical and pedagogical needs at the beginning and expected learner skills at the end, respectively. The *language*, the *instructional level* (*beginner*, *intermediate*, *expert*), *first/second-level categories*, and *tags* are listed. Each course has only one first-level category and one second-level category, while tags can be more than two for the same course. Other course fields identify the current *price* and the *discount*. Statistical attributes list the *estimated course duration* in hours. Finally, some boolean flags indicate *certification release* and *lifetime access availability*. The *Curriculum* entity includes a hierarchical list depicting the *chapters* and their *lessons*.

Due to privacy constraints, the *Instructor* and *Learner* entities only include information available on the corresponding public profiles. Each entity instance is uniquely identified by a fake *id*, so that the id stored into the dataset does not correspond to the real id of the user. Each instructor is described by the job title and biography. Each learner has a flag indicating whether the profile is public.

The COCO strength is the large amount of relationships among primitive entities. In *Teach*, the pairs of *instructor id* and *course id* model the association among instructors and the courses they teach. One instructor can teach more than one course and the same course can have one or more instructors. Then, each pair of *course id* and *learner id* in *Participate* defines the courses that the learner has been attending. In *Wish*, the id pairs set the courses each learner has inserted into the wish-list. Finally, *Evaluate* contains *learner id* and *course id* together with the [0–5] *rating* with step 0.5, the *comment* and the *timestamp*.

2.4 Statistics

Describing COCO in numbers, it includes 43,113 courses distributed into a taxonomy composed of 15 first-level categories and 133 second-level categories, as reported in Fig. 2(a, b). The courses distribution is unbalanced for both first-level categories (avg. 2874; st.dev. 2334; min 477; max 7985) and second-level categories (avg. 324; st.dev. 475; min 7; max 3196). Similarly, the languages distribution along courses follows such trend, as depicted in Fig. 2(c). The languages employed in at least 25 courses are explicitly named. Only 21% of courses do not use English as primary language. Regarding the courses structure, each course contains 43 lessons in average (st.dev. 46; min 1; max 863).

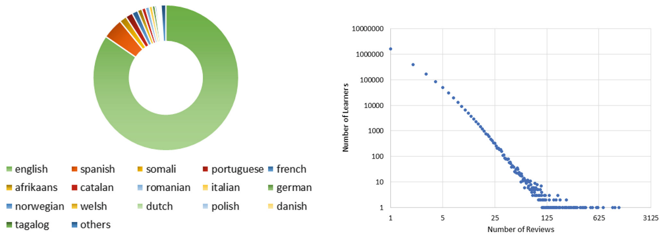


(a) (b)

(c) (d)

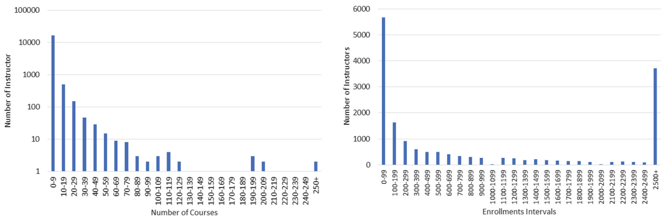
Fig. 2. The distribution of courses per (a) first-level category, (b) second-level category, (c) content language, (d) number of learners reviews.

In COCO, there are 2,546,865 learners who provided 4,584,313 ratings and 2,453,865 comments. The sparsity of the rating matrix is 0.99583%. Only learners with at least one rating are included, while each course can have zero or more ratings. In Fig. 2(d), the distribution of the number of ratings per courses (avg. 119; st.dev. 812; min. 1; max. 57,346) shows a downward trend, but there is a large number of courses with a lot of ratings. Figure 3 shows the distribution of learners per (a) the review language and (b) the number of provided ratings. Despite some learners have made a lot of reviews, the average number of ratings per learner is low (avg 2; st.dev. 3; min 0; max 1159). COCO also incorporates 16,963 instructors. Figure 4(a) shows their distribution based on the number of courses they teach (avg. 3; st.dev. 10; max 748; min 1). In Fig. 4(b), the distribution of instructors according to the number of learners enrolled into their courses appears divided in two blocks, with a peak of the number of instructors with few enrolled students (avg. 4,036; st.dev. 19,238; min 1; max 850,496).



(a) (b)

Fig. 3. The distribution of learners per (a) review language, (b) number of reviews.



(a) (b)

Fig. 4. The distribution of instructors per (a) courses and (b) learners they manage.

3 Experimental Use Cases

This section depicts two potential use cases made possible by having COCO and a set of experiments to demonstrate how they are as promising as challenging.

3.1 Multi-class Content-Based Course Classification

Multi-class classification assigns each course to one category chosen among a set of different options in a pre-defined taxonomy. E-learning domain is semantically challenging and hardly leverages several services that other domains have already exploited. The automated classification makes easier both the categorization and the exploration of courses. Given a set of training course records $D = \{d_1, ..., d_n\}$ such that each one of them is labeled with a category c_i of a set $C = \{c_1, ..., c_m\}$, multi-class classification is a supervised task aimed to infer a model $f : D \longrightarrow C$ that relates each course record in D to a category in C. Then, the model is able to predict the category for a course whose category is unknown. For the evaluation, we included the most successful algorithms, namely Support Vector Machine (SVM), Decision Trees (DT), Random Forest (RF) and Naive Bayes (NB) [10]. Their implementation was provided by the Scikit-learn library.

First, minority categories were over-sampled to account for unbalanced categories. Then, the evaluation protocol worked as follows. For each setting, we adopted 10-fold stratified cross-validation, maintaining the original category distribution in training and test sets. For each fold, the algorithms were fed with each features set extracted from each course attribute in red in Fig. 1. Then, the performance was evaluated using weighted precision (W-P), recall (W-R) and f-measure (W-F1). Each metric was first calculated for each category, and the average is found, weighted by the number of true instances for each category.

Table 1 reports the best results that were obtained in the considered experimental settings. The results provide empirical evidence that the TF-IDF generally produces better results than high level features. The poor performances of the latter ones could indicate that they are not suited to capture the fine-grained information. Although they clearly capture the most relevant characteristics of each course at the human eyes, the results are not the same when they are analyzed by machines; therefore, advanced semantic enriching models and techniques need to be studied to get meaningful insights from semantic information.

Table 1. The best classification results with first-level category as target.

Source attribute	Algorithm	Features type	W-P	W-R	W-F1
Long Description	*SVM*	*Nouns-TF-IDF*	*0.79*	*0.78*	*0.77*
Short Description	SVM	TF-IDF	0.61	0.61	0.60
Objectives	SVM+SGD	TF-IDF	0.74	0.73	0.72
Requirements	NB	Nouns-TF-IDF	0.57	0.47	0.43

3.2 Course Recommendation

Recommender systems can be one of the solutions proposed to navigate among the overwhelming alternative courses. We considered the sets of users U, courses C, ratings R in the range [0–5], supposing that no more than one rating can

Table 2. The rating prediction performance measured with Root Mean Square Error.

Metric	NP	SVD	SVD++	NMF	Slope one	Co-clustering
RMSE	1.051	0.7796	*0.7755*	0.8334	0.8582	0.9595

be made by any user for a given item, writing this rating as r_{ui}. The most popular task in recommender systems is predicting the rating score. The goal is to learn a model $f : U \longrightarrow C$ that predicts the rating f_{ui} of a user u for a new item i. For the evaluation, we employed Normal Predictor (NP) as baseline, K-Nearest-Neighbor (KNN), K-Nearest-Neighbor with K-Means (KNN-K), K-Nearest-Neighbor with Z-Score, Matrix Factorization SVD and the SVD++, and Non-negative Matrix Factorization. Their implementation is in Surprise[10]. The ratings R are divided into a training set R_{train} used to learn f and a test set R_{test} to evaluate prediction accuracy with Root Mean Squared Error (RMSE).

The average RMSEs for various algorithms with their default parameters on a 5-folds cross-validation procedure are depicted in Table 2. The folds were the same for all the algorithms and the random seed was set to 0. The results highlight that SVD++ performs the best among all the investigated scenarios. Its performance is significantly better than the baseline Normal Predictor. SVD++ shows an improvement of about 0.28 on RMSE compared to such baseline. However, the results still need to be improved; in this direction, advanced semantic enriching techniques which leverage content-based course information in addition to the ratings can be a viable solution to get better prediction results.

4 Other Existing Datasets

Other datasets have been frequently used in technology-enhanced learning. They differ from COCO in terms of size and shape, domain, and context of user interaction. Here, we discuss only the most prominent alternatives.

The Dataset of Joint Educational Entities (DAJEE) [11] includes about 20 K resources extracted from 407 online courses distributed in 10 first-level categories and 36 second-level categories. Over 484 academic instructors are mentioned. However, the authors built an ontology aimed to detect patterns in academic teaching. No interaction among learners and courses is listed together.

The Technology Entertainment Design (TED) dataset [12] contains around 1 K talks and 69 K users who made more than 100 K ratings and 200 K comments. In addition to the difference in scale, this collection embraces only resources and no educational information such as course requirements and objectives is given.

Multimedia Education Resource for Learning and Online Teaching (MERLOT) [13] is a collection of free and open online resources contributed by an international education community. It includes over 40 K materials with 19 different material type categories. It incorporates a variety of resource types larger

[10] http://surpriselib.com/.

than COCO, although they were collected from face-to-face learning lessons, and no feedback on learners' preferences and interactions is included.

The HarvardX-MITx Person-Course Dataset [14] includes interactions in 17 MITx and HarvardX courses on edX platform. These data are aggregated records representing individual activities in one course. They combine several learner information (e.g. *degree*, *gender*, *birth date*) and provide data on interactions within courses (e.g. *number of viewed activities*, *number of published posts*). The data granularity and the number of courses limit the applicable analysis methods.

The Metadata for Architectural Contents in Europe (MACE) dataset [15] provides metadata-based access to learning resources in repositories all over Europe. It offers access to about 150,000 learning objects, holding together about 47,000 tags, 12,000 classification terms and 19,000 competency values. COCO outruns it in scale and presence of learner feedback, while including a smaller number of classification terms and tags.

Comparing to these datasets, COCO advantages are its scale, completeness and comprehensiveness, so it can be employed for wider scenarios and use cases.

5 Conclusions

In this paper, we presented COCO, a complete and comprehensive online course collection enriched with stakeholder interactions crawled from Udemy. It presently refers to more than 43 K online courses, 16 K instructors and 2,5 M learners who have provided 4,5 M reviews. COCO provides data about courses, learners and instructors, including enrollments, reviews, and wish-lists. Furthermore, we proposed possible use cases supporting online course delivering. The experiments demonstrated that such use cases are challenging and need novel research to manage online courses proliferation. Advanced semantic-based techniques can extract insightful information to support stakeholders during organization and delivery. Moreover, COCO is expected to support reproducible evaluation in other technology-enhanced learning approaches.

We will keep maintaining and updating this dataset in terms of resources, learner and instructor attributes, and interactions within courses, extending them with data coming from other online course providers.

Acknowledgments. Danilo Dessì and Mirko Marras acknowledge Sardinia Regional Government for the financial support of their PhD scholarship (P.O.R. Sardegna F.S.E. Operational Programme of the Autonomous Region of Sardinia, European Social Fund 2014–2020, Axis III "Education and Training", Specific Goal 10.5).

References

1. eLearning Market Trends and Forecast 2017–2021 (2017). https://www.docebo.com/resource/elearning-market-trends-and-forecast-2017-2021/. Accessed 20 Nov 2017
2. G2crowd grid for online course providers (2017). https://www.g2crowd.com/categories/online-course-providers. Accessed 20 Nov 2017

3. Basu, S., Yu, Y., Zimmermann, R.: Fuzzy clustering of lecture videos based on topic modeling. In: 2016 14th International Workshop on Content-Based Multimedia Indexing (CBMI), pp. 1–6. IEEE (2016)
4. Dessì, D., Fenu, G., Marras, M., Recupero, D.R.: Leveraging cognitive computing for multi-class classification of e-learning videos. In: European Semantic Web Conference, pp. 21–25. Springer (2017)
5. Yang, H., Meinel, C.: Content lecture video retrieval using speech and video text information. IEEE Trans. Learn. Technol. **7**(2), 142–154 (2014)
6. Drachsler, H., Verbert, K., Santos, O.C., Manouselis, N.: Panorama of recommender systems to support learning. In: Recommender Systems Handbook, pp. 421–451. Springer (2015)
7. Fazeli, S., Rajabi, E., Lezcano, L., Drachsler, H., Sloep, P.: Supporting users of open online courses with recommendations: an algorithmic study. In: Advanced Learning Technologies (ICALT), pp. 423–427. IEEE (2016)
8. Class central. https://www.class-central.com/. Accessed 20 Nov 2017
9. Course talk. https://www.coursetalk.com. Accessed 20 Nov 2017
10. Aggarwal, C.C., Zhai, C.: A survey of text classification algorithms. In: Mining Text Data, pp. 163–222 (2012)
11. Estivill-Castro, V., Limongelli, C., Lombardi, M., Marani, A.: DAJEE: a dataset of joint educational entities for information retrieval in technology enhanced learning. In: Proceedings of the 39th International ACM SIGIR, pp. 681–684. ACM (2016)
12. Pappas, N., Popescu-Belis, A.: Combining content with user preferences for ted lecture recommendation. In: 2013 11th International Workshop on Content-Based Multimedia Indexing (CBMI), pp. 47–52. IEEE (2013)
13. Merlot (2017). https://www.merlot.org/merlot/index.htm. Accessed 20 Nov 2017
14. Ho, A.D., Reich, J., Nesterko, S.O., Seaton, D.T., Mullaney, T., Waldo, J., Chuang, I.: Harvardx and mitx: the first year of open online courses (2014)
15. Mace. http://ea-tel.eu/tel-research/mace/. Accessed 20 Nov 2017