

# *SeqSense*: video recommendation using topic sequence mining

Chidansh Bhatt, Matthew Cooper, and Jian Zhao

FX Palo Alto Laboratory, Palo Alto, CA 94304 USA,  
{bhatt, cooper, zhao}@fxpal.com,  
<http://www.fxpal.com>

**Abstract.** This paper examines content-based recommendation in domains exhibiting sequential topical structure. An example is educational video, including Massive Open Online Courses (MOOCs) in which knowledge builds within and across courses. Conventional content-based or collaborative filtering recommendation methods do not exploit courses' sequential nature. We describe a system for video recommendation that combines topic-based video representation with sequential pattern mining of inter-topic relationships. Unsupervised topic modeling provides a scalable and domain-independent representation. We mine inter-topic relationships from manually constructed syllabi that instructors provide to guide students through their courses. This approach also allows the inclusion of multi-video sequences among the recommendation results. Integrating the resulting sequential information with content-level similarity provides relevant as well as diversified recommendations. Quantitative evaluation indicates that the proposed system, *SeqSense*, recommends fewer redundant videos than baseline methods, and instead emphasizes results consistent with mined topic transitions.

**Keywords:** content-based video recommendation, educational video

## 1 Introduction

The explosive growth of expository content in the modern video sharing era is well documented. This content ranges from one-off how-to videos to professionally produced, multi-course certification programs. One prominent example is the Massive Open Online Course (MOOC) videos distributed on platforms such as Coursera, edX, and Udacity.

A course may consist of 30-90 videos each targeting specific learning objectives. The videos are sequenced by the instructor in a syllabus to facilitate students' comprehension. Syllabi are frequently partitioned hierarchically into sections containing subsets of videos covering closely related concepts. The syllabi thus encode sequential relationships between specific videos. Because the videos themselves are often short and topically coherent, the syllabus also reflects inter-topic sequential relationships.

However, the syllabus remains a prescribed "one size fits all" approach. Its inability to accommodate the diversity of learner profiles is cited as a factor

contributing to low MOOC student retention rates [1], [2], [3]. One approach to this challenge is to aggregate content *across* related courses available on multiple platforms and use recommendation to enable users to flexibly navigate the expanded collection of videos. Recommendations drawn from multiple courses can provide additional perspectives on concepts of interest.

Recently, MOOC providers have recognized that their consumers include many professionals or “lifelong learners” who are no longer students [4]. These learners often consume online videos to achieve professional and career growth rather than complete a certification or degree program [5].

In our view, these learners will require more flexible access to a broader range of content. We hypothesize that these users will seek finer grained information closer to the individual video level than the course level. As well, we believe these users will benefit from the ability to choose among a set of related content across multiple courses to best address their information needs.

We propose recommendation methods that aim to better support this broader set of users, and first aggregate data from various MOOC platforms. We build a common topic-based representation of the course content based on text transcripts. Next, we identify sequential relationships between videos’ topics across the corpus. For this, we use the most prominent topics detected per video and the (partial) ordering of videos in the syllabi as input to a sequence mining module. The output is two sets of significant inter-topic transitions observed in the course syllabi. Finally, we re-rank recommendations generated by a traditional text processing pipeline using the sequential information. Our experiments show the resulting recommendations originate from a more diverse set of courses, and also better reflect inter-topic orderings in the course syllabi.

## 2 Related work

This approach builds on two areas of related work. The first is topic modeling of text document collections. Other works have used Latent Dirichlet Allocation (LDA) [6] for video recommendation, including [7]. In social media applications, temporal variants of LDA have been proposed such as Wang *et al.* [8]. On-line LDA [9] proceeds by working with a partitioned corpus. However, application of these methods require a global ordering across the corpus such as time. There is no natural sequencing of content across independently created courses that are organized conceptually. Seeded LDA [10] has shown promise in applications with educational data, but requires data labeling. [11] is a system for educational video aggregation which focused on improved access to content by searching text displayed within videos, but did consider sequential relationships between videos.

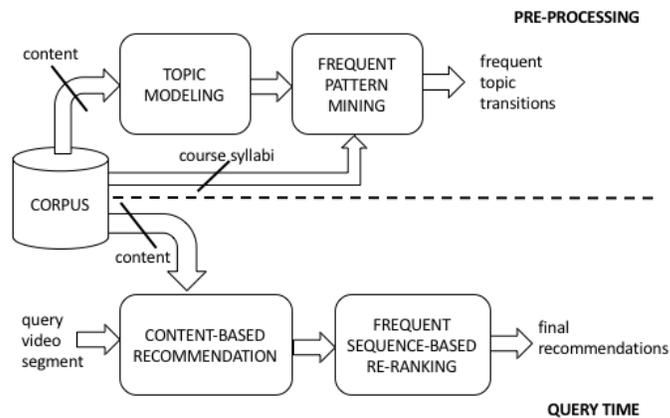
Similarly, many applications of sequence mining are distinct from video recommendation. Kinnebrew *et al.* [12] studies sequence mining approaches to modeling student behavior, but this is not focused on modeling instructional content. [13] argues that topic mining alone is insufficient for linking web videos to supplement digital textbooks. We augment course data with corresponding syllabi. More recently, [14] studied the role of topic sequencing in student performance in

the context of a tutoring system. Their analysis indicates that the topic sequencing can critically impact student performance. Thus, the sequential organization of topics by experts within course syllabi may provide valuable information for improving recommendation. [15] facilitates information discovery in educational hypermedia systems by applying sequential pattern mining to user logs, but not the content. The system of [16] enables content adaptation based on user modeling within a single course. Incorporating user modeling is an important direction in which we plan to extend the work presented here.

We combine established topic modeling and sequential pattern mining methods for educational content recommendation. Sequence learning can be achieved with several methods including Markov models, recurrent neural networks (RNNs), and sequential pattern mining (SPM). We select sequential pattern mining here as it typically requires less parameter tuning or training data [17]. We apply the Top- $K$  sequential pattern mining (TKS) [18] and the Top- $K$  non-redundant sequential rules (TNS) algorithms [19] to discover important sequential rules within our syllabus database. The discovered rules indicate which specific course topics exhibit sequential relationships across our collected courses.

### 3 System description

We integrate automatically mined inter-topic relationships into a conventional content-based recommendation engine using re-ranking. The data flow appears in Figure 1. Pre-processing is illustrated above the dashed line. These steps are performed off-line and updated when the corpus changes. The lower portion shows processing at recommendation time.



**Fig. 1.** Data flow. Offline pre-processing is depicted above the dashed line. Processing at recommendation time appears below it.

### 3.1 Data pre-processing

We assemble content from a number of courses, including videos, available text transcripts, and other meta-data. The courses are designed from varied perspectives by each instructor. Generating recommendations across the breadth of content requires a common data representation. Useful visual attributes such as whether the format is a classroom lecture, khan-academy style electronic ink, or Coursera style slide-based videos, etc. are largely captured by associated meta-data. Most platforms provide semantically rich closed caption transcripts enabling the identification of videos’ prominent topics.

Explicit topics within videos are difficult to align across courses due to differences in vocabulary and the contexts in which material is presented. Also, any manual mapping is not scalable or generalizable to alternate domains. We thus discover latent topics present across the collection using LDA [6], an established unsupervised topic modeling method. In the LDA model, each document is generated by a mixture of a low dimensional set of topics. We estimate the appropriate number of topics [20] for the corpus and perform visualization to verify topic quality [21]. Each video is modeled by its distribution over the discrete set of  $Z$  latent topics with  $Z = 30$  throughout.

### 3.2 Sequential pattern mining

Sequential pattern mining (SPM) algorithms identify prominent subsequences within a sequence database [22]. Our database contains sets of topics corresponding to each video and the partial orderings derived from each course syllabus. Detected subsequences are associated with support measures indicating their prevalence within the database.

Our aim is to use the currently watched video to recommend videos covering concepts users are likely to watch next. However, frequent topic subsequence detection alone is not sufficient for prediction. Sequential rule mining addresses prediction by discovering rules of the form  $X \Rightarrow Y$ , where  $X$  and  $Y$  are two sets of topics.  $X \Rightarrow Y$  denotes the rule “topic(s)  $Y$  appear in the sequence after topic(s)  $X$ ”. Intuitively, the support of frequent patterns corresponds to the marginal probability of the subsequence, whereas sequential rules have corresponding confidence measures analogous to *conditional* probabilities  $P(Y|X)$ . For additional details see [19].

Denote the *topic signature* of the  $i^{th}$  video by  $V_i = \{k : P^{(i)}(z_k) \geq 0.1\}$  which is the discrete set of topic indices weighted at least 0.1 in the video’s topic distribution. We construct an ordered sequence of video topic signatures according to each course’s syllabus. These sequences are aggregated into a sequence database. Course syllabi often employ a hierarchical structure in which related videos are grouped into course sections. Both the sections and the videos within a section are ordered. In syllabi with multiple hierarchical levels, we use the level immediately above individual videos for section grouping.

The Top- $K$  non-redundant sequential patterns (TNS) algorithm [19] detects sequential rules that reflect *global* analysis of inter-topic transitions. TNS eliminates rules that are deemed “redundant” (rules that are implied by other rules

having the same support and confidence) to capture more varied sequences and automatically fine-tunes the minimum support parameter.

Some topic sets observed in our videos are relatively infrequent and will be overlooked in the global analysis. The Top- $K$  sequential pattern mining (TKS) algorithm [18] finds sequential patterns within a given minimum and maximum length such that a set of items must appear within a defined allowed gap. For *local* analysis, we collect the sequences that include each video’s signature. We apply TKS with the constraint that each video’s topic set appears within a distance of 3 to 6 in the sequence. We then apply TNS algorithm on these derived sequences to find significant sequential rules within this local data subset. By this design, each video’s topic signature is described by sequential rules in the local analysis. Each sequential rule in the local analysis has a corresponding confidence score. This application of sequential pattern mining produces sets of prominent topic transitions describing the sequence database both globally and locally.

### 3.3 Recommendation

In the scenario of interest, a user watches a video which comprises the query to the system. The first step depicted below the dotted line in Figure 1 is to issue the query against our baseline content-based recommendation system. This currently uses standard tf/idf vectorspace retrieval [23] based on the video transcripts with ranking according to cosine similarity. The initial results emphasize videos with similar content. These videos are appropriate in cases when users wish to augment their understanding of a topic. However, when users are ready to advance to related topics, topic-transition knowledge can enhance recommendation. Denote the latent topic distribution of the video with index  $r$  by  $P^{(r)}$ . We have examined a variety of scoring criteria detailed below:

**Topic similarity score (TS):** The signatures of the query video  $V_q$ , and recommended video  $V_r$ , are matched in terms of overlap and probability values. The TS score combines the Jaccard similarity between the signatures, and the number of probabilities for common topics that are within a threshold difference of 0.2.

$$\text{Sim}_{\text{TS}}(V_q, V_r) = \text{Jaccard}(V_q, V_r) + \frac{1}{Z} \sum_{z=1}^Z \delta(|P^{(q)}(z) - P^{(r)}(z)| \leq 0.2) . \quad (1)$$

**Global sequence score (GS):** We retrieve  $N$  support and confidence score values,  $\{(s_n, c_n)\}$  from the mined *global* sequential patterns with antecedent values matching  $V_q$  and consequent values matching subsets of  $V_r$ . The GS score is

$$\text{Sim}_{\text{GS}}(V_q, V_r) = \frac{1}{N} \sum_{n=0}^N c_n \frac{y_n}{|V_r|} + \frac{s_n}{D_G} . \quad (2)$$

$D_G$  is the total number of global sequences, and  $y_n$  is the length of the matched subset of  $V_r$ . This score emphasizes results consistent with topic transitions mined in the global analysis of the corpus.

**Local sequence score (LS):** We retrieve  $M$  additional support and confidence score values from mined *local* sequential patterns with antecedent matching a subset of  $V_q$  and consequent matching a subset of  $V_r$ . The LS score is

$$Sim_{LS}(V_q, V_r) = \frac{1}{M} \sum_{m=0}^M c_m \frac{y_m}{|V_r|} + \frac{s_m}{D_q}. \quad (3)$$

$D_q$  is total number of mined local sequences with antecedent matching any subset of  $V_q$ , and  $y_m$  is the length of the matched subset of  $V_r$ . The LS score preferentially weights results with topics that appear in close proximity in the sequence database to the specific topics in  $V_q$ . We apply feature scaling to bring all scores into the range  $[0, 1]$  before linearly fusing the scores with uniform weights in the experiments.

### 3.4 Sub-sequence recommendations

Most recommendation systems generate lists of highly ranked results. For MOOC data, it is useful to consider the context of videos (i.e., adjacent videos in the syllabus). We observe that many of the top recommendation results are in close proximity within their original course sequences. We thus generate sub-sequence results with videos in sequential order subject to constraints including total subsequence duration, proximity in the original sequence (allowable gaps), and sequence length. We consider the top 20 results and group videos by their original courses and sort these videos based on their sequence lengths. We generate sub-sequences of length 2-3 from any sequences longer than 4. The sub-sequences are ordered according to their course syllabus. The sub-sequence similarity score is the per-video average of the constituent similarity scores. Finally, we filter the results to retain only the highest scoring subsequence from each course.

## 4 Evaluation

### 4.1 Data

We report pilot experiments using a *pilot* corpus of five MOOCs related to machine learning, as shown in Table 1. We performed LDA with  $Z = 30$  latent dimensions to represent the content. For sequence analysis, the parameter  $K = 8,000$  for TNS on the pilot database with a minimum confidence value of 10%. TNS on the 53 sequence pilot database generated 62,688 maximum sequential pattern candidates with minimum support of 2 from which we retain the top 8,000 rules. For the local rules, the number of latent topics associated with each video signature is much smaller (avg. 3.27), and we thus set  $K = 100$  to avoid extraneous patterns.

Additionally, we extend the experiments to a larger corpus with 4,186 videos from 42 courses from additional MOOC platforms including edX and Udacity. When TNS is run on the full database with 537 sequences, it generated 237,846

**Table 1.** Five MOOC courses on various aspects of machine learning comprising the *pilot* experimental corpus.

<i>MOOC Title</i>	<i>Videos</i>	<i>Sections</i>
Machine Learning	112	41
Neural Nets for Machine Learning	78	16
Machine Learning With Big Data	28	14
Machine Learning: Regression	120	44
Machine Learning Foundations	117	35
<b>Total</b>	455	150

maximum sequential pattern candidates with minimum support of 20% from which we retain the top 8,000 rules. For local rules, the number of latent topics associated with each video signature is on average 4.1, and we again set  $K = 100$ .

## 4.2 Metrics

We validate the recommendation system in the absence of user evaluation by contrasting its recommendations with those produced by a conventional text-based pipeline. We employ several measures of the topical characteristics of the highest ranked recommendations, and also use of the original syllabus containing the query video. We denote the video index that follows the query with index  $q$  in its original syllabus by the index  $q_{-next}$ .

**Matching topics:** indicates the average number of the top ten results with signatures  $V_r$  overlapping with  $V_q$  or  $V_{q_{-next}}$ . Ideally, we want the recommendation to be relevant to both the query video and the next video to smoothly traverse through related content. Vectorspace retrieval provides more granular, word-based similarity while LDA can provide topic similarity.

**Coverage of topics:** indicates the average number of topics from  $V_q$  appearing in at least one of the top ten results. Here, we assume that the topics that appear in the sequel video, but not in the query video are desirable for recommendation. We denote this topic set  $V_{q_{-next}.d} = V_{q_{-next}} \setminus V_q$  to assess how much any conceptual gap between  $V_q$  and  $V_{q_{-next}}$  can be bridged by recommended videos.

**Novelty within section:** measures the redundancy of recommendations between query videos belonging to the same section in a syllabus. We believe that emphasizing topic transitions in our ranking can account for finer grained differences between related queries. Our intuition is that as users move through proximate videos in a course syllabus, generated recommendations that include the same results repeatedly limit users' ability to expand their understanding.

**Diversity of courses:** provides statistics on the average number of distinct courses from which the top ten recommendations originate. We assume that different courses are prepared with different objectives, viewpoints, and constraints (e.g. experience, time). Thus, recommendations spanning different sources can provide a more flexible user experience. Overall, these metrics enable comparing the proposed system incorporating sequential information with the content-based recommender in terms of relevance, redundancy, and source diversity.

**Table 2.** Summary comparing several system variants in terms of Matching, Coverage, Diversity and Novelty as described in Section 4.2.

Pilot Results							
Ranking	Matching		Coverage			Diversity	Novelty
	$V_q$	$V_{q-nxt}$	$V_q$	$V_{q-nxt}$	$V_{q-nxt-d}$		
CB	7	7	0.6	0.6	0.5	1	4
GS	7	7	0.6	0.6	0.5	2	9
LS	8	8	0.6	0.6	0.4	2	8
GS + LS	8	8	0.6	0.6	0.4	2	9
TS + GS + LS	9	8	0.7	0.6	0.5	2	9

Complete Results							
Ranking	Matching		Coverage			Diversity	Novelty
	$V_q$	$V_{q-nxt}$	$V_q$	$V_{q-nxt}$	$V_{q-nxt-d}$		
CB	9	9	0.6	0.7	0.4	1	7
GS	10	9	0.7	0.7	0.4	2	9
LS	9	9	0.5	0.5	0.2	2	8
GS + LS	9	9	0.5	0.6	0.3	2	9
TS + GS + LS	10	9	0.8	0.7	0.4	2	9

### 4.3 Quantitative Results

The experiments compare content-based (CB) recommendations to variants of our system which use re-ranking functions described in Section 3. Results from the pilot dataset appear in the upper portion of Table 2 and the bottom portion shows results for the complete dataset. The CB recommendations concentrate within the query video’s course based on the distinctiveness of each instructor’s word usage. Using CB on the pilot dataset we found that the first recommendation from any different course appears on average at rank position twelve on average. To provide more diverse recommendations, we filter out results from the same course as  $V_q$  for this evaluation and assess the the ability of CB and the proposed methods to direct users to related content from other courses.

Table 2 shows that the matching and coverage measures for all methods are comparable with some differences between the pilot dataset and the complete dataset. GS and LS perform similarly in the pilot study, but with the extended dataset, GS outperforms LS. In terms of the diversity and novelty measures, the integration of sequence mining shows a clear impact for both datasets. The novelty measure examines recommendation results within sections of the courses. Using the CB baseline on the pilot dataset, the sets of the top ten results for videos within a section produced an average overlap of 6 (stdev=3.7) or novelty of 4. Using the sequence based scoring functions GS and LS, the average overlap for the recommendations was 1 (stdev=3.8) as in Table 2. Using the complete dataset, the CB recommender produces an average overlap of 3 (stdev=3.9) compared to sequence based scoring (TS + GS + LS) with the average overlap for the recommendations 1 (stdev=3.0) or novelty of 9. Thus, the proposed approach

**Table 3.** Example recommendation results comparing **CB** (left) and **GS + LS** (right). The query video is “Unsupervised learning introduction” with  $V_q = [4\ 15\ 16\ 21]$  and sequel video “k-means algorithm” with  $V_{q-next} = [7\ 9\ 15\ 21]$ .

rank	CB		GS + LS	
	video title	$V_r$	video title	$V_r$
1	clustering documents an unsupervised learning task	[2 10 15 22]	spark mllib clustering	[9 21 22]
2	k means a clustering algorithm	[15 21 22]	the k-means clustering algorithm	[9 13 15 17 21]
3	other examples of clustering	[15 21 22 28]	spark mllib frequent item associated sets	[0 9 21]
4	from weighted k-nn to kernel regression	[2 13 21 22]	loading exploring wikipedia data	[16 21 22]
5	weighted k nearest neighbors	[2 15 22]	unsupervised learning with clustering	[9 10 15 21]

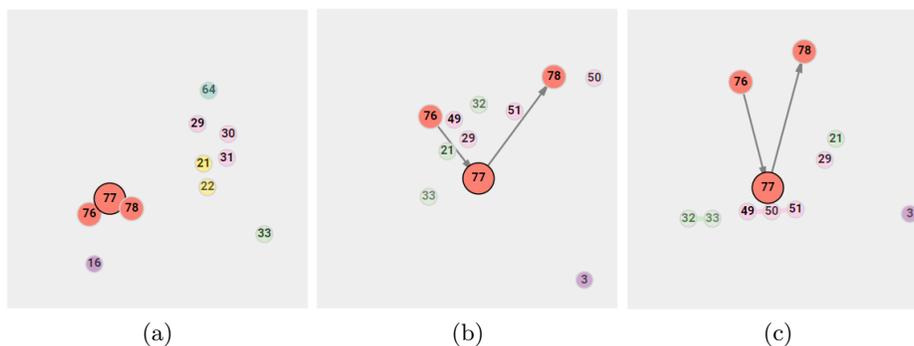
uses the sequential relationships to provide more diverse recommendations while preserving essential similarity to the query.

#### 4.4 Example Results

Table 3 compares recommendation results using the pilot dataset from CB with those from GS+LS for the example query video “Unsupervised learning introduction” with  $V_q = [4\ 15\ 16\ 21]$  from the “Machine Learning” MOOC. The sequel video in the syllabus is “k-means algorithm” with  $V_{q-next} = [7\ 9\ 15\ 21]$ . CB provides two recommendations (ranks 4 and 5) unrelated to unsupervised learning. The re-ranked results span varied aspects (e.g., algorithms, tools, applications) of unsupervised learning from more courses. Also, GS+LS shows greater topic matching and coverage between the query, sequel, and recommended videos.

We also use multidimensional scaling (MDS) to further examine our recommendations. MDS [24] is a standard dimension reduction technique that uses proximity to communicate similarity in an abstract low dimensional space. We use it here to visualize the relationships between recommended results. We use the TS+GS+LS similarity scores between the query video, the top ten recommendations, and the adjacent (previous and next) videos to the query from its course syllabus. We convert these to dissimilarity values by negation. MDS takes this set of dissimilarities and returns a layout of points corresponding to each data instance such that the distances between the points spatially reflect the videos’ pairwise relationships.

Figure 2 shows example MDS layouts of recommendation results for the query video for lecture 77 in the 112 video course titled “Machine learning”. The query title is “K means algorithm.” The color code indicates the different courses from which the individual videos originate. Arrows connect lectures according to the course syllabus of the query. We include the neighboring videos in the syllabus



**Fig. 2.** MDS visualizations of recommendations using CB scores in panel (a), TS+GS+LS scores in panel (b) and sub-sequence recommendations based on TS+GS+LS scores in panel (c). Each circle represents a video labeled with its lecture sequence number. Circles with the same color indicate videos in the same course.

(i.e., lectures 76 “Unsupervised learning introduction” and 78 “Optimization objective”).

Panel (a) shows the MDS visualization of the CB recommendations, and the videos from the query’s course are closely clustered. The recommended videos are also clustered but the layout doesn’t readily reveal any relationship between the recommendations and the query or its neighbors. Panel (b) shows the MDS layout using TS+GS+LS scores. The layout suggests stronger relationships between the recommendations and the query context. Specifically, various recommendations show more relative similarity to the previous, query, or subsequent videos. This context is not evident in the layout of the CB recommendations. This visualization suggests the various recommendations can address a broader range of user information needs. Panel (c) shows the MDS layout of the sub-sequence recommendations using TS+GS+LS scores. Also, when we convert the recommendation list to the sub-sequences it is interesting to observe that sub-sequence with lecture ids [49, 50: “Estimating parameters from soft assignment”, 51] is rated highest in Panel (b). Both that sub-sequence and the sub-sequence [29: “Hope for unsupervised learning” , 30: “the K means algorithm”, 31: “k means as coordinate”] generated from CB recommendations are from the course “Machine learning clustering and retrieval”. The CB results represents the preliminary part of the course whereas the TS+GS+LS sub-sequence occurs later in that course and is more relevant to the discussion of clustering in the query video from the course “Machine learning”. Considering sub-sequences provides flexibility in navigating the recommendations and can use more complete context across multiple adjacent videos.

## 5 Conclusion

We presented a recommendation system, SeqSense, that exploits sequential relationships between topics. These inter-topic relations occur commonly in educational content as well as corpora documenting structured, ordered processes. Our initial experiments indicate SeqSense makes fewer redundant recommendations than a conventional content-based method. Instead, recommendations reflect topic transitions observed in course syllabi created by the instructors. This provides a natural means by which users can explore collections of related videos originating from independent, often siloed, content platforms. Recommendations optionally include short sub-sequences of videos to more completely convey content beyond video-to-video level recommendation. The use of a global topic model and sequence-based recommendation can enhance the learning experience by allowing users to more flexibly navigate videos across independent siloed platforms while avoiding redundant conceptual content.

While collaborative filtering based recommendation systems process user behavioral data to drive recommendation, many scenarios like education may not readily share user data outside their institutions. However, user logs or models remain a powerful complementary source of information to improve the proposed system. More generally, the query can be expanded to integrate viewing history information, facets based on meta-data, or extracted keyphrases [25]. Beyond incorporating auxiliary data, our future work continues on several fronts. We are designing an interface to enable users to playback content and explore our corpus via recommendation and so that we can perform a more complete study of the system's utility. This work will also involve further visualization of recommendation results to facilitate flexible exploration and navigation in the larger corpus.

## References

1. Onah, D.F., Sinclair, J., Boyatt, R.: Dropout rates of massive open online courses: behavioural patterns. *EDULEARN14 Proceedings* (2014) 5825–5834
2. Brief, U.I.f.L.L.U.P.: Making large-scale literacy campaigns and programmes work. (2016)
3. Cope, B., Kalantzis, M.: *e-Learning Ecologies: Principles for New Learning and Assessment*. Routledge (2017)
4. Levin, R.: Announcing Coursera for Business. <https://blog.coursera.org/announcing-coursera-for-business/> (2016) [Online; accessed 19-July-2017].
5. Zheng, S., Rosson, M.B., Shih, P.C., Carroll, J.M.: Understanding student motivation, behaviors and perceptions in moocs. In: *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing. CSCW '15, New York, NY, USA, ACM* (2015) 1882–1895
6. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. *Journal of machine Learning research* **3**(Jan) (2003) 993–1022
7. Zhu, Q., Shyu, M.L., Wang, H.: Videotopic: Content-based video recommendation using a topic model. In: *ISM, IEEE Computer Society* (2013) 219–222

8. Wang, Y., Agichtein, E., Benzi, M.: Tm-lda: efficient online modeling of latent topic transitions in social media. In: KDD, ACM (2012) 123–131
9. AlSumait, L., Barbará, D., Domeniconi, C.: On-line lda: Adaptive topic models for mining text streams with applications to topic detection and tracking. In: ICDM, IEEE (2008) 3–12
10. Ramesh, A., Goldwasser, D., Huang, B., Iii, H.D., Getoor, L.: Understanding mooc discussion forums using seeded lda (2014)
11. Adcock, J., Cooper, M., Denoue, L., Pirsiavash, H., Rowe, L.A.: Talkminer: a lecture webcast search engine. In: ACM Multimedia, ACM (2010) 241–250
12. Kinnebrew, J.S., Loretz, K.M., Biswas, G.: A contextualized, differential sequence mining method to derive students’ learning behavior patterns. *Journal of Educational Data Mining* **5**(1) (2013)
13. Agrawal, R., Christoforaki, M., Gollapudi, S., Kannan, A., Kenthapadi, K., Swaminathan, A.: Mining videos from the web for electronic textbooks. In: International Conference on Formal Concept Analysis, Springer (2014) 219–234
14. Doroudi, S., Holstein, K., Alevan, V., Brunskill, E.: Sequence matters, but how exactly? a method for evaluating activity sequences from data. In: International Conference on Educational Data Mining EDM. (2016)
15. Morales, C.R., Pérez, A.P., Soto, S.V., Martínez, C.H., Zafra, A.: Using sequential pattern mining for links recommendation in adaptive hypermedia educational systems. *Current Developments in Technology-Assisted Education* **2** (2006)
16. Pardos, Z.A., Tang, S., Davis, D., Le, C.V.: Enabling real-time adaptivity in moocs with a personalized next-step recommendation framework. In: ACM Conference on Learning @ Scale, ACM (2017) 23–32
17. Fournier-Viger, P., Faghihi, U., Nkambou, R., Nguifo, E.M.: Cmrules: Mining sequential rules common to several sequences. *Knowledge-Based Systems* (2012)
18. Fournier-Viger, P., Gomariz, A., Gueniche, T., Mwamikazi, E., Thomas, R.: Tks: Efficient mining of top-k sequential patterns. In: ADMA (1). Volume 8346 of Lecture Notes in Computer Science., Springer (2013) 109–120
19. Fournier-Viger, P., Tseng, V.S.: Tns: mining top-k non-redundant sequential rules. In: SAC, ACM (2013) 164–166
20. Guille, A., Soriano-Morales, E.P.: Tom: A library for topic modeling and browsing.
21. Sievert, C., Shirley, K.E.: Ldavis: A method for visualizing and interpreting topics. In: Proceedings of the workshop on interactive language learning, visualization, and interfaces. (2014) 63–70
22. Fournier-Viger, P., Lin, J.C.W., Kiran, R.U., Koh, Y.S.: A survey of sequential pattern mining. *Data Science and Pattern Recognition* (2017) 54–77
23. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press (2008)
24. Kruskal, J.B.: Multidimensional scaling by optimizing goodness of fit to a non-metric hypothesis. *Psychometrika* **29**(1) (1964) 1–27
25. Chuang, J., Manning, C.D., Heer, J.: ”without the clutter of unimportant words”: Descriptive keyphrases for text visualization. *ACM Trans. Comput.-Hum. Interact.* **19**(3) (2012) 19