Towards a custom designed mechanism for indexing and retrieving video transcripts

Gabriel Turcu[1], Stella Heras[2], Javier Palanca[2], Vicente Julian[2], and Marian Cristian Mihaescu[1]

[1] Faculty of Automatics, Computers and Electronics, Craiova, Romania
gabriel.turcu97@gmail.com, cmihaescu@software.ucv.ro
[2] Departamento de Sistemas Informáticos y Computación, Universitat Politècnica de València, Spain
{sheras,jpalanca,vinglada}@dsic.upv.es

Abstract. Finding appropriate e-Learning resources within a repository of videos represents a critical aspect for students. Given that transcripts are available for the entire set of videos the problem reduces to obtaining a ranked list of video transcripts for a particular query. The paper presents a custom approach for searching the 16.012 available video transcripts from https://media.upv.es/ at Universitat Politècnica de València. An inherent difficulty of the problem comes from the fact that transcripts are in the Spanish language. The proposed solution embeds all the transcripts using feed-forward Neural-Net Language Models, clusters the embedded transcripts and builds a Latent Dirichlet Allocation (LDA) model for each cluster. We can then process a new query and find the transcripts that have the LDA results closest to the LDA results for our query.

Keywords: Latent Dirichlet Allocation · NNLM word embeddings · Clustering

1 Introduction

Searching for appropriate e-Learning resources (i.e., videos, quizzes, presentations, etc.) is one of the most critical activities for students that are willing to improve their knowledge. General purpose search engines may do an excellent job, but custom designed professional search tools are more advisable for better results. Thus, within the area of e-Learning, the Video Base Learning (VBL) [19] stands a particular place which gets more and more attention due to its effectiveness in teaching and learning. The significant technological advances have given the VBL a vital role in improving learning outcomes and properly designing VBL environments.

One critical aspect in VBL is the retrieval of relevant videos given an input query. This problem has been addressed in [3] by reviewing a wide range

of machine learning algorithms that have been used for indexing and retrieving learning materials. Among the most utilised indexing parameters, there are the ones that refer to text coming from natural language, documents or images. The various formats under which text may be shaped are web document, logs, XML, structured or semi-structured data. The general picture is completed by a wide range of indexing algorithms such as clustering, Ant colony, semantic index, SemTree, text index tree, B-Tree, etc.

The most critical shortcoming of search systems is that they are general and for existing implementations, they highly depend upon underlying data. Therefore, we have developed a specific search system over a dataset of video transcripts from a Spanish public university. A particularity of our input dataset consists of the fact that their transcripts are in the Spanish language. This poses new challenges as few developments that were done for the English language were also implemented for the Spanish language.

This paper presents a custom designed mechanism for indexing and retrieving video transcripts. The task is to index available video transcripts such that for an input query provided by a user the retrieval mechanism should return a list with the most representative videos. In our particular case, the input is represented by a large set of educational video transcripts and the search is accomplished by learners who are seeking learning materials.

The proposed approach takes as input 16.012 available video transcripts and builds a dataset by extracting necessary features. The dataset consists of a bag-of-words (BoW) and its corresponding matrix of NNLM embedding results. Given K, representing the number of domains which span the video transcripts, we run a clustering algorithm to obtain a partitioning. Thus, available video transcripts are assigned into clusters in an attempt to group by instances (i.e., video transcripts) into domains. Once the domains are obtained, we further run a LDA [6] algorithm to get a list of topics and their score. Then, given a query, we determine the closest centroid and therefore obtain the domain of the query to which it belongs. Finally, by searching into the acquired domain's instances, we get a ranked list of video transcripts that are closest to the query and return them to the user. Preliminary validation of the proposed solution has been performed manually by comparison with the outputs provided by the currently existing search mechanism.

The paper has the following structure. Section 2 presents related work about several indexing and retrieval systems and in particular the contexts in which specific algorithms have been used. Section 3 shows the proposed design of the data analysis workflow. Section 4 presents experimental results with emphasis on each processing step and especially on comparative analysis with the currently existing search mechanism. Finally, section 5 presents the conclusions and future works.

2 Related Work

In recent years, the rise of Massive Online Open Courses (MOOCs), and Technology Enhanced Learning (TEL) systems, in general, has highlighted even more the importance of having efficient and accurate information retrieval systems. This boom of online multimedia content has also brought an information overload problem. It is useless to have a large amount of online educational resources if students are not able to quickly find those that best suit their educational needs and preferences.

Information retrieval is a vast area of research with a large number of contributions in different domains [2]. Among them, many methods and algorithms have been proposed to find and retrieve textual and multimedia content from the web. In the latter case, multimedia IR encompasses different tasks, such as feature extraction and indexing from different types of sources (video frames, images, audio tracks, speech transcripts, etc.). There are several approaches proposed in the literature for multimedia IR: content-based IR; audio and music retrieval; speech recognition; or retrieving and browsing video.

In *content-based* multimedia IR, the primary objective is to identify and extract features related to image contents. Following this approach, in [7], authors compare 'traditional' engineered (hand-crafted) features (or descriptors) and learned features for content-based semantic indexing of video documents. Learned (or semantic) features are obtained by training classifiers in the context of the TRECVID[3] semantic indexing task.

In [9], authors propose a combination of content-based video indexing approaches: text-based, feature-based, and semantic-based. The text-based approach focuses on using keywords or tags to describe video content. The feature-based approach aims to extract low-level features such as colour, texture, shape and motion from the video data and use them as indexing keys. The semantic-based approach focuses on the automatic video content annotation with their semantic meanings.

Following an approach similar to the proposed in this paper, the work presented in [10] highlights the shortcomings of current multimedia indexing and retrieval techniques, mainly based on sparse tagging, and deals with content-based video indexing and retrieval using an LDA probabilistic framework [6].

In our same domain of lecture video retrieval, [15] presents an indexing method for recorded videos of computer science courses. This proposal uses the automatic transcriptions from a speech-recognition engine to create a chain index for detailed browsing inside a lecture video. Also, in [18], authors presented a method for content-based video indexing and retrieval in sizable German lecture video archives. This paper applies a combination of

[3] `https://www-nlpir.nist.gov/projects/tv2018/`

automatic video segmentation and key-frame detection with a technique to extract textual meta-data by applying video Optical Character Recognition (OCR) technology on key-frames and Automatic Speech Recognition (ASR) on lecture audio tracks. Applying a technique based on multi-modal language models for lecture video retrieval, in [8] authors demonstrated that this method outperforms LDA-based methods when speech transcripts are error-free, but the model shows similar performance for noisy text.

Other works use BoW based methods to classify and retrieve videos (with keywords [4], subjects [17] or visual features [12]). However, BOW based models cannot describe the content of an image objectively and neglect the spatial distribution of visual words and the order of words in transcripts.

Although multimedia retrieval based on speech transcripts may seem very similar to text retrieval, in practice, it is very different. For instance, speech transcripts lack from or have many flaws detecting structure (punctuation, paragraphs). Therefore, the appropriate proposals for feature extraction and indexing of written text are not always the best to perform this task on speech transcripts. The same happens with the approaches that use textual metadata or OCR to analyse lecture slides and get keywords from them. For our system, we take as input Spanish speech transcripts from an extensive database of lecture recordings from many types of university courses. Therefore we focus on the problem of *feature extraction and video indexing from speech transcripts* (which may result in noisy text gathered from automatic speech recognition engines) to deliver the most suitable set of videos for a specific keyword search.

3 Proposed Approach

Searching within a large available set of videos represents a challenging task for a student. The current video search mechanism implemented in the multimedia platform *https://media.upv.es/* performs a full-text search over the title and keyword fields of the videos. It examines the words stored in such fields and tries to match with the search query made by the user. These full-search techniques may suffer from the lack of semantics and context on the search since they only take into account the word included in the query as it is. In the approach proposed in this work, we present a procedure that is much more context-aware, being able to classify videos according to their content using their transcripts.

3.1 Outline of Data Analysis Pipeline

1. **Build the bag-of-words.** As the video transcripts represent the primary input for the data analysis process, the first step is to build a BoW by tokenizing the transcripts to remove stop-words and words that are too

short. Once the BoW has been created the next task is to determine the domains in which the transcripts may group. As no labels are being provided, the most effective solution is to implement an unsupervised learning algorithm (i.e., clustering) for grouping the transcripts. The most significant limitation of this approach is represented by the fact that the BoW cannot be used directly as an input to a clustering algorithm and why we have to embed the transcripts using state of the art feed-forward neural net language models.
Another issue may regard the fact that we are dealing with a large number of words from BoW and with a reasonably large number of transcripts, which will end up in having a sufficiently large input dataset for clustering. From the application domain perspective, a cluster should group transcripts that belong to a particular domain. By observing the university's media website, we have identified four domains: Arts & Humanities, Engineering & Architecture, Sciences (Biological Sciences), and Social & Legal.

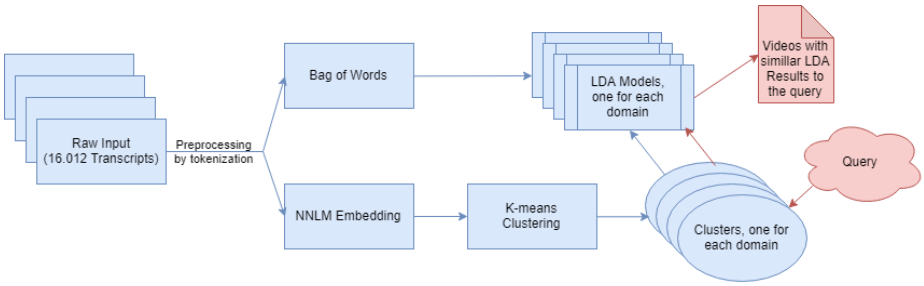

Fig. 1. Data analysis pipeline-cool

2. **Compute matrix with NNLM Word Embedding Results.** We want to divide the transcripts into four different domains and to do that we have to run a clustering algorithm on the transcripts, but since we can't run K-means on string objects, we have to transform our transcripts into a data format that our clustering algorithm understands. To accomplish this, we use the NNLM word embeddings. NNLM word embedding saves a lot of space by learning a distributed representation for words which allows each training sentence to inform the model about an exponential number of semantically neighbouring sentences. The model learns simultaneously a distributed representation for each word along with the probability function for word sequences, expressed in terms of these representations. The generalisation is obtained because a sequence of words that have never been seen before gets high probability if it is

made of words that are similar (in the sense of having a nearby representation) to words forming an already seen sentence [5]. The output from the NNLM embedding is a vector of 128 float numbers for each transcript, which amounts in total to a float matrix that is 16.012 lines (one for each transcript) and 128 columns. The output vector forms the input for our clustering algorithm.

3. **Build clusters of transcripts.** The next step consists of running a clustering method for assigning items (i.e., transcripts) into groups. From this perspective, each obtained cluster represents a domain within the entire set of available videos. The primary purpose of the clustering algorithm is to bring together items (i.e., transcripts) for which the terms have similar NNLM embedding results over the entire transcripts corpus. One option is to use a standard simple k-means clustering algorithm [11] and provide a value for K as a domain knowledge person provides or use other algorithms that do not require for the number of clusters, such as xMeans [14]. Finding the optimal number of clusters in a particular dataset represents itself a challenging research issue and is not covered by the current works. Another parameter that needs setup within the clustering process is the distance function (i.e., Euclidean, Jaccard, cosine, edit, etc.). Taking into account that the embedding results represent the input data, the current approach uses the Euclidean distance along with standard clustering quality metrics: SSE, homogeneity, completeness, Adjusted Rand-Index or Silhouette coefficient.

4. **Model each cluster and find its topics and scores.** For each cluster of transcripts (i.e., BoW corpus of that particular cluster and the dictionary of that cluster) we use LDA to determine its topics and associated scores. From an LDA perspective, the *words* are represented by the tokens obtained in the first step, the *documents* are represented by the transcripts and the *corpus* is represented by the set of transcripts for a particular cluster. As output, LDA determines the *latent topics* and their characterization in terms of *words*. Intuitively, for each cluster (i.e., domain) the LDA model creates a list of topics defined by scores whose values add up to one. More, each topic is defined by a list of words with their coefficients. Both in the case of the topic's scores and coefficient's values, the interpretation is that a more significant number represents a more important topic or word. Finally, the model (i.e., the topic's scores and their list of coefficients and words) is serialized for later querying.

5. **Query and retrieve a ranked list of results.** Once a query is obtained from a user we need to find the cluster/domain to which it belongs. Since the query is regarded as a transcript, it is firstly preprocessed and its cluster is being determined. Determining the cluster to which the query belongs needs computing the NNLM embedding results of the words

that make up the query. The embedding values for the query are being computed by considering the query as a transcript.
Once the embedding results from the query are determined, the closest centroid of already built clusters determines the cluster to which the query is assigned and therefore the LDA model to be queried.
For each transcript from the assigned cluster, we compute the difference in topics between the query's LDA results and transcript's LDA results. A lower score indicates a smaller difference and therefore a transcript that matches the query better. Thus, once LDA provides the topics and scores for each cluster/domain, we may end up obtaining a ranked list of transcripts (i.e., movies) which are most similar to the query. We then take the top 5 transcripts and recommend those.

4 Experimental Results

4.1 Input Dataset

The input dataset consists of 16.012 video transcripts that are accessible through a *json* file. The structure and raw data for a record from the *json* file are presented in the following example.

```
{"video": {
  "_id": "00054a38-5a32-4db2-ae9c-85c296015c3b",
  "hidden": "False",
  "title": "Programa Mathematica gratis y online",
  "type": "polimedia",
  "mimetype": "video/mp4",
  "width": "640",
  "height": "480",
  "src": "politube.mp4",
  "slides": {
    "menuitem": [
      {"mimetype":"image/jpg","url":"frame.0.jpg","time":"0"},
      {"mimetype":"image/jpg","url":"frame.48.jpg","time":"48"},
      {"mimetype":"image/jpg","url":"frame.431.jpg","time":"431"
          }
    ]
   "duration": "564.523537",
   "transcription": "...very long text... (or empty)"
  }
}}
```

This dataset comes from an online e-learning platform (i.e., https://media.upv.es) from Universitat Politecnica de Valencia (UPV). The UPV's platform has

recording facilities to create educational videos (most of them in Spanish) which are finally publicly available as MOOCs.

The dataset includes the information necessary to process the video transcription and also includes additional information such as the identifiers needed to download the video (such as the `_id` field, the `src` field or the `type` field). Similarly, the fields `mimetype`, `width` and `height` are the meta-information of the video file. Information such as the title of the video or its duration in seconds is also available. However, something that characterizes these videos, which are educational videos recorded in a room dedicated to this activity, is that they are accompanied by information related to the slide presentation if used during the recording of the video.

During the recording of the videos a software that detects the change of slide in the screen of the presenter was used. Thus, each slide detected in the dataset was captured and tagged. Therefore, the `slides` field contains a list of identified slides with their filename to be downloaded, the mime-type of the file and the instant (in seconds) of the video in which the slide change was detected.

What makes this dataset interesting is that it not only includes a collection of videos with a specific theme (educational videos), but also includes the transcription of the audio of these videos, and even each of the slides that have been used in the presentation, along with its temporary label. This approach enables a much more in-depth analysis of this data set.

4.2 Numerical Results

Each transcript from the raw input dataset has been transformed into a BoW for which NNLM values were computed and saved into a matrix. The dimension of the matrix is 16.012 (i.e., number of transcripts) x 128. What the embedding algorithm does is it maps from text to 128-dimensional embedding vectors.

Then, the embedding matrix has been used as input for simple k-means algorithms from Scikit-learn [13] to obtain a distribution of items (i.e., transcripts) into four clusters. For the clustering process, the items are represented by transcripts, features are represented words, and embedding results represent the values that build up the input dataset. Table 1 presents several BoW samples along with computed embedding results and their corresponding CA (cluster assignments).

The running of simple K-means algorithm provides following distribution of transcripts into clusters: 22.339%, 32.466%, 13.343% and 31.852%. Once we have determined the four clusters of transcripts, we further run the LDA algorithm to determine a lexical model of each cluster as a list of topics along with their scores and a list of words that make up that topic. The obtained model also consists of computed scores for each word within the topic. Table 2 presents sample numerical results for the transcripts described in table 1.

Table 1. Sample BoW with embedding results and cluster assignments.

ID	Sample BoW	Sample Embedding Results	CA
3	hola vamos a ver la parte cinco de la documentacion del software basicamente seria otro programa que nos quedaba veamos por explicar seria el cloc...	0.8655922, 0.77715206, -0.16605175, -1.6408461, -0.8250744, 0.02193201, 0.700683, 0.7213742, -0.3716459, 1.5170424, 0.13685325, -0.20364942, 0.5022141, ...	1
4	hola vamos a ver ahora la parte politica de calidad es este digamos que el objeto que se realizaria simplifica de tareas basicamente estos se utiliza mucho...	0.65827113, 0.5848848, 0.12338758, -1.5416939, -0.59042096, -0.17713968, 0.78873384, 0.5304664, -0.37286106, 1.0820895, 0.5255295, -0.27624443, 0.50664973, 0.15184559, ...	1
6	bienvenidos y bienvenidas a esta unidad de formacion en la cual trataremos sobre los usos de la letra cursiva somos sepalo assange del servicio de promocion y normalizacion...	0.33565775, 0.48877674, 0.17549452, -1.6629573, -0.91474277, 0.3601234, 0.68089503, 0.55267304, -0.40141803, 1.3872166, 0.2669923, -0.25762805, 0.7387372, 0.17457546, ...	3

For each transcript presented in table 2 the computed topic scores have the sum equal to 1. This approach makes interpretation straightforward in the way that a topic score of 0.844 is a big score indicating that the topic represented is an excellent representative for that transcript. Further, each topic is represented by a list of words and their coefficients. In the same line of approach, larger value in coefficient is an indication of a more critical word among the words that make up the topic.

Once the query is being obtained from the user, it is preprocessed and embedding results are computed. At this stage, the *corpus* is represented by the available transcripts along with the query. Thus, the query is reduced to an array of words (i.e., a BoW corpus) which is assigned to the closest cluster (i.e., nearest centroid). Determining the cluster (i.e., the domain) to which the query belongs opens the way to further investigating its associated LDA model.

Table 3 presents the query results: the query, the assigned cluster (i.e., the domain), the LDA scores (i.e., the topic IDs and its score) and the ranked results (transcript ID and score). The results are ranked by the computed score from the fourth column as this score represents the difference between the query's LDA score and transcript's LDA score. A lower value in the transcript's LDA score represents a smaller difference; therefore, a transcript that is a better match for the query.

4.3 Validation

Validation is the part of the project that has proved to be the most difficult mostly because we are working with real-life data and therefore we do not

Table 2. Sample LDA models with their topics and scores

ID	LDA results: topics and scores
3	**Score**: 0.8441817164421082 **Topic**: 0.005*"filtr" + 0.004*"estad" + 0.002*"clas" + 0.002*"tension" + 0.002*"senal"
	Score: 0.0837591215968132 **Topic**: 0.002*"dat" + 0.002*"registr" + 0.001*"formulari" + 0.001*"eolic" + 0.001*"electr"
	Score: 0.06878719478845596 **Topic**: 0.001*"datagr" + 0.001*"estil" + 0.001*"motor" + 0.001*"dataset" + 0.001*"wrait"
4	**Score**: 0.6999539732933044 **Topic**: 0.005*"filtr" + 0.004*"estad" + 0.002*"clas" + 0.002*"tension" + 0.002*"seÃśal"
	Score:0.24506276845932007 **Topic**:0.002*"dat" + 0.002*"registr" + 0.001*"formulari" + 0.001*"eolic" + 0.001*"electr"
	Score: 0.053080759942531586 **Topic**: 0.001*"datagr" + 0.001*"estil" + 0.001*"motor" + 0.001*"dataset" + 0.001*"wrait"
6	**Score**: 0.6111074686050415 **Topic**: 0.001*"oracion" + 0.001*"pronombr" + 0.001*"agu" + 0.001*"sistem" + 0.001*"instal"
	Score: 0.22719042003154755 **Topic**: 0.001*"plan" + 0.001*"edifici" + 0.001*"sistem" + 0.001*"element" + 0.001*"derech"
	Score: 0.049981363117694855 **Topic**: 0.001*"plan" + 0.001*"control" + 0.001*"derech" + 0.001*"punt" + 0.001*"nod"
	Score: 0.04900844022631645 **Topic**: 0.001*"fibr" + 0.001*"atom" + 0.001*"molecul" + 0.001*"temperatur" + 0.001*"sistem"
	Score: 0.04536473751068115 **Topic**: 0.001*"arrend" + 0.001*"derech" + 0.001*"pacient" + 0.001*"valencian" + 0.001*"anten"
	Score: 0.010526408441364765 **Topic**: 0.001*"sistem" + 0.001*"pec" + 0.001*"ulcer" + 0.001*"presion" + 0.001*"aliment"

Table 3. Query results.

Query	CA	LDA scores (topic ID and score)	Ranked list of transcripts (transcript ID and score)
Ciencias de la Computacion	2	(0,0.4422385) (1,0.053041767) (2,0.04431011) (6,0.033507172) (7,0.42583835)	(41107,0.10977402180433274) (765,0.11200470328330994) (18460,0.11249668002128602) (3895,0.11415494084358216) (1236,0.11463153958320618)
aprendizaje automatico	2	(9,0.6999294) (8,0.033338577) (7,0.03334638)	(29906,3.75695526599e-05) (16256,4.62792813777e-05) (35066,5.43296337155e-05) (3212,6.037577986724e-05) (17793,0.0001398846507487)
permanente para la proteccion de los animales en cria instituido	2	(6,0.014290362) (7,0.8714059) (8,0.014287368) (9,0.014287801)	(44794,1.2902542948724e-05) (27277,1.7145648598676e-05) (44466,0.03214275650680065) (26721,0.04286431334912777) (41793,0.04287060722708702)

possess the ground truth needed so that we may validate our work with ease. For validation, a couple of approaches have been investigated and tested including:

- Manual validation by looking at the results from a query which represent the top transcripts whose LDA scores are the closest to our query's LDA scores and seeing if the transcripts' content reflect the query.
- Applying TextRazor's state-of-the-art Natural Language Processing and Artificial Intelligence API [16] to parse, analyze and extract semantic meta-data from the transcripts to see if the detected Categories and Topics are related to our query.
- We applied the LSTM Siamese Text Similarity [1] algorithm on the top 3 results from a couple of queries to check if the recommended transcripts for those queries are similar to each other. We had to use the English transcripts because there wasn't anything already trained for the Spanish language. We downloaded the transcripts that were translated from Spanish to English and applied the algorithm. If they would have been marked as similar by the LSTM Siamese algorithm and if the categories and topics from TextRazor would be relatable to our queries, that would tell us that our program does a good job at recommending transcripts based on the content in them. The results from the LSTM Siamese algorithm were inconclusive, however. This happened because the LSTM Siamese algorithm works best if the sentences are short and if they retain a meaning while our transcripts were detected imperfectly by the speech recognition engine after which they were imperfectly translated into English and there was also the issue that the transcripts are very long with the average number of words in a transcript being a bit over 1000 words.

5 Conclusions and Future Works

In this paper, we have implemented a custom procedure for indexing and retrieving video transcripts. The input transcripts are from educational videos available from *https://media.upv.es/*. The data analysis pipeline consists of stemming, computing NNLM embedding results, clustering transcripts and building one LDA model for each transcript. Once the LDA models are available, the query provided by the user is preprocessed and labelled (i.e., assigned to a cluster) and the corresponding LDA model is used for obtaining the most similar topics (i.e., with highest scores) and therefore the most significant words with their coefficients. These results trace back to the originating transcripts, and consequently, a ranked list of videos is obtained as output.

One of the critical limitations of the current approach is that it uses a fixed number (i.e., four) of clusters/domains. This approach is due to prac-

tical reasons since we do not have any ground truth indicating the exact number of clusters that are in the 16.012 transcripts. Finding the correct number of clusters from the dataset may bring significant improvements regarding the relevance of the obtained ranked list. Besides, a future goal is to improve the current search mechanism which takes into consideration only the title of the video.

Another shortcoming is that no ground-truth data is available and no mechanism to monitor learner's search queries or search truth judgments. Coping with these shortcomings may end up in building a recommender system that takes into account the context of the learner and previously trained classifiers.

References

1. Aman Srivastava: LSTM Siamese Text Similarity. https://github.com/amansrivastava17/lstm-siamese-text-similarity (Apr. 2019)
2. Baeza-Yates, R., Ribeiro, B.d.A.N., et al.: Modern information retrieval. New York: ACM Press; Harlow, England: Addison-Wesley, (2011)
3. Bakar, Z.A., Kassim, M., Sahroni, M.N., Anuar, N.: A survey: Framework to develop retrieval algorithms of indexing techniques on learning material. Journal of Telecommunication, Electronic and Computer Engineering (JTEC) **9**(2-5), 43–46 (2017)
4. Basu, S., Yu, Y., Singh, V.K., Zimmermann, R.: Videopedia: Lecture video recommendation for educational blogs using topic modeling. In: International Conference on Multimedia Modeling. pp. 238–250. Springer (2016)
5. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. Journal of machine learning research **3**(Feb), 1137–1155 (2003)
6. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of machine Learning research **3**(Jan), 993–1022 (2003)
7. Budnik, M., Gutierrez-Gomez, E.L., Safadi, B., Quénot, G.: Learned features versus engineered features for semantic video indexing. In: Content-Based Multimedia Indexing (CBMI), 2015 13th International Workshop on. pp. 1–6. IEEE (2015)
8. Chen, H., Cooper, M., Joshi, D., Girod, B.: Multi-modal language models for lecture video retrieval. In: Proceedings of the 22nd ACM international conference on Multimedia. pp. 1081–1084. ACM (2014)
9. Elleuch, N., Ammar, A.B., Alimi, A.M.: A generic framework for semantic video indexing based on visual concepts/contexts detection. Multimedia Tools and Applications **74**(4), 1397–1421 (2015)
10. Iyer, R.R., Parekh, S., Mohandoss, V., Ramsurat, A., Raj, B., Singh, R.: Content-based video indexing and retrieval using corr-lda. arXiv preprint arXiv:1602.08581 (2016)
11. MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. vol. 1, pp. 281–297. Oakland, CA, USA (1967)

12. Ngo, C.W., Jiang, Y.G., Wei, X.Y., Wang, F., Zhao, W., Tan, H.K., Wu, X.: Experimenting vireo-374: Bag-of-visual-words and visual-based ontology for semantic video indexing and search. In: TRECVID (2007)
13. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. Journal of machine learning research **12**(Oct), 2825–2830 (2011)
14. Pelleg, D., Moore, A.W., et al.: X-means: Extending k-means with efficient estimation of the number of clusters. In: Icml. vol. 1, pp. 727–734 (2000)
15. Repp, S., Grob, A., Meinel, C.: Browsing within lecture videos based on the chain index of speech transcription. IEEE Transactions on learning technologies **1**(3), 145–156 (2008)
16. Toby Crayston: The Natural Language Processing API. `https://www.textrazor.com/technology` (Apr. 2019)
17. Van Nguyen, N., Coustaty, M., Ogier, J.M.: Multi-modal and cross-modal for lecture videos retrieval. In: Pattern Recognition (ICPR), 2014 22nd International Conference on. pp. 2667–2672. IEEE (2014)
18. Yang, H., Meinel, C.: Content based lecture video retrieval using speech and video text information. IEEE Transactions on Learning Technologies (2), 142–154 (2014)
19. Yousef, A.M.F., Chatti, M.A., Schroeder, U.: Video-based learning: a critical analysis of the research published in 2003-2013 and future visions (2014)