

Videopedia: Lecture Video Recommendation for Educational Blogs Using Topic Modeling

Subhasree Basu¹, Yi Yu², Vivek K. Singh³, and Roger Zimmermann¹

¹ National University of Singapore, Singapore 117417, Singapore
{sbasu, rogerz}@comp.nus.edu.sg

² National Institute of Informatics, Tokyo 101-8430, Japan
{yiyu@nii.ac.jp}

³ University of South Florida, Tampa FL 33620, USA
vivek4@mail.usf.edu

Abstract. Two main sources of educational material for online learning are e-learning blogs like Wikipedia, Edublogs, etc., and online videos hosted on various sites like YouTube, Videolectures.net, etc. Students would benefit if both the text and videos are presented to them in an integrated platform. As the two types of systems are separately designed, the major challenge in leveraging both sources is how to obtain video materials, which are relevant to an e-learning blog. We aim to build a system that seamlessly integrates both the text-based blogs and online videos and recommends relevant videos for explaining the concepts given in a blog. Our algorithm uses content extracted from video transcripts generated by closed captions. We use topic modeling to map videos and blogs in the common semantic space of topics. After matching videos and blogs in the space of topics, videos with high similarity values are recommended for the blogs. The initial results are plausible and confirm the effectiveness of the proposed scheme.

1 Introduction

The lecture videos captured in classrooms contain a substantial portion of the instructional content [12]. These videos have several advantages over graphic and textual media, e.g., portrayal of concepts involving motion, the alteration of space and time, the observation of dangerous processes in a safe environment, etc [14]. They also provide the viewer with the benefit of pausing and reviewing at leisure, which is not possible in traditional classrooms. These videos are often hosted on the intranet of the universities and also on numerous online sites like Coursera⁴, Khan Academy⁵, etc. Leading universities like MIT and Stanford have made their lectures available online for distance learning. Besides, there are numerous lecture videos uploaded to video sharing sites like YouTube⁶, Videolectures.net⁷, etc., to facilitate further learning. With the increase in Internet speed globally, the demand for multimedia content has been increasing over the past couple of years, with a projected Internet video consumption of 56,800 PB/month in 2018 [1]. E-learning and smart learning videos constitute a huge fraction of this consumption. Hence there is an increase in the demand and consumption of multimedia content related to e-learning.

⁴ <https://www.coursera.org/>

⁵ <https://www.khanacademy.org/>

⁶ <https://www.youtube.com/>

⁷ <http://videolectures.net/>

The entire e-learning process is not limited to videos alone. Text based blogs and websites like Wikipedia⁸ and Edublogs⁹ are also important sources of learning material. Wikipedia is a user generated knowledge base available on the Internet that is frequently visited for learning and reference by users from both academia and industry. In essence, wikis offer an online space for collaborative authorship and writing. A survey demonstrates that students reported that they frequently, if not always, consult Wikipedia at some point during their course-related research [8]. Wikipedia serves as a convenient go-to source when students are stuck on some concepts and can not move forward [8]. However, Wikipedia usually only provides a brief explanation of concepts without the detailed descriptions. To provide comprehensive learning for students over the Internet, the contents from multiple sources, e.g., text and videos are required and should be provided in an integrated platform [15]. For example, it will be more effective to enhance the Wikipedia contents with multimedia information such as videos that cover some online lectures about the topic described in the wikipedia.

In this paper, we design a system for integrating the blogs and the videos containing educational multimedia data and propose a novel recommendation system called *Videopedia*. We provide the necessary matching between the texts in the webpages and the lecture videos based on their contents. Retrieving suitable videos for a particular webpage is challenging mainly because webpages and videos are represented in different forms and finding a correlation between them is not straightforward. Also e-learning lecture videos mostly consist of lecture slides, whiteboard/blackboard content and sometimes the instructors delivering the lecture. Hence extracting the content of these videos based on visual words is not a plausible solution. Video transcripts have been generated using automatic speech recognition (ASR) of the audio as an attempt for content detection [16]. Most of the extracted texts by ASR contain errors due to poor audio recording quality, which degrades the spoken text's usability for video retrieval [6]. Recently, the majority of video sharing platforms like Ted¹⁰, YouTube, etc., provide videos with automatic video transcripts. These are manually created closed captions (CC) and contain less error than ASR. These transcripts provide the content of the lecture videos which we can then mine to find the correlation with the text-based blogs.

Our novel algorithm implemented as a part of *Videopedia* thus employs techniques of topic modeling in extracting the contents of the webpages and the videos. We use Latent Dirichlet Allocation (LDA) [3] to find the topics in a particular website and those in a particular video. These topics provide better representation of the content of the video compared to video metadata. The topics for the videos are extracted from video transcripts, stored on the server and indexed by their topic distributions. The topics representing the contents of a blog are generated at run-time. *Videopedia* determines the similarity between the topics of a webpage and those of a video and recommends the relevant videos based on the similarity. In summary, our specific contributions in this paper are as follows:

⁸ <http://www.wikipedia.org/>

⁹ <https://edublogs.org/>

¹⁰ <http://www.ted.com/>

- For a webpage recommend the relevant videos that would elaborate the concepts introduced in the webpage.
- Provide a comprehensive framework which combines multimodal data, i.e., text and video for an enhanced e-learning experience.
- Effectively use video transcripts for lecture video recommendation.

The rest of the paper is organized as follows. In Section 2 we provide a brief summary of the related work. In Section 3 we give a description of the model of our system and Section 4 summarizes our experiments and the results. We conclude with the direction of our future work in Section 5.

2 Related Work

Chen et al. attempted to link web-videos to wikipages by leveraging Wikipedia categories (WikiCs) and content-duplicated open resources (CDORs) [5]. Later they adopted a multiple tag property exploration (mTagPE) approach to hyperlink videos to webpages [6]. Okuoka et al. have used Wikipedia entries to label news videos [15]. This labeling was primarily done by using date information and labels that are extended along the topic thread structure following a time-series semantic structure. Roy et al. have used Wikipedia articles to model the topic of a news story [17]. Topic modeling has found its use in a lot of research areas to find the latent topics in documents, many of which being in the e-learning scenario. Wang and Blei combined probabilistic topic modeling with collaborative filtering to recommend scientific papers [19]. Chen, Cooper, Joshi, and Girod have proposed a Multi-modal Language Model (MLM) that uses latent variable techniques to explore the co-occurrence relation between multimodal data [4]. They mainly use MLMs to index text from slides and speech in lecture videos. One of the major contributions to indexing and searching lecture videos has been done by Adcock et al. [2]. They have designed a system that provides a keyword-based search on a database of 62,406 video lectures and talks. Zhu, Shyu and Wang used topic modeling techniques in video recommendation systems and designed VideoTopic [20]. The recommendation system generates a personalized video recommendation list to fit the user's interests. The bag-of-words model used in VideoTopic contains both visual as well as textual features.

In contrast, *Videopedia* is designed to automatically recommend relevant videos based on the content of the webpages and not for a particular user. The aim of our system is to integrate e-learning materials from different online sources and not to develop a latent variable model as MLM [4]. One of the main problems in extracting the content of the lecture videos is the absence of meaningful visual words. Hence we need to extract the content covered in the speech of the lecture videos. Compared to Chen et al. [5], our system extracts the educational material covered in the videos and not just metadata. Also the videos in *Videopedia* are indexed by the latent topics present in them and not timing information as is done by Okuoka et al [15]. We are recommending lecture videos for Wikipedia entries and not using the content of wikipages for mining topics in our videos like Roy et al [17]. In the domain of online education, topic modeling has been used only for mining the text and not utilized on lecture video content analysis. Hence ours is a first attempt in that direction. Also by indexing the videos by topics, we propose a content based search and not just keyword based search as proposed by Adcock et al. [2].

3 System Model

Videopedia proposes a novel video search technique based on topic extraction and automatically recommends videos based on the content of a particular webpage or educational blog. The existing e-learning systems provide keyword-based video search like Videotopic [2] and hence they can not be used to automatically recommend videos for wikipages and other blogs. The webpage and the video represent two different media formats and finding a correlation between them is the main challenge in designing such a system. An important factor in our recommendation system is that we are recommending lecture videos and hence mining the visual content of the videos extracted through visual words do not provide too much semantic information. This is because the lecture content is delivered either through slides or blackboard/whiteboards and the visual content does not change much throughout the entire video. However, the instructor explains most of the points written in the slides and/or on boards, and the speech of the videos will represent the content of these videos. Therefore, we use the transcripts of these videos and use them as the representative of the content of these lecture videos.

Fig. 1 provides a pictorial description of the proposed model. A user views the webpage (e.g., a Wikipedia page) via a browser. The request for videos comes from the webpage on the browser by sending the URL of this webpage to the server. The content of the webpage is extracted and then the latent topics are generated from the content. The video repository is stored on the server and the topic models of these videos are generated beforehand. The matching between the topic models of the webpage and those of the video transcripts in the repository is also done on the server side in real time. The recommended videos are then linked to the webpage on the browser.

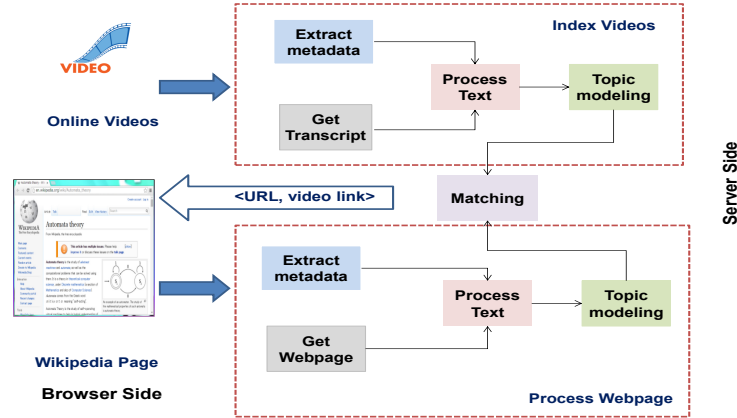


Fig. 1: System model for Videopedia

3.1 Dataset Used for Recommendation

For this research, we collected the videos from the YouTube channel for National Programme on Technology Enhanced Learning (NPTEL) and also from Videlectures.net.

Our video repository now consists of 3,000 videos covering the following subjects – Humanities and Social Sciences, Metallurgical and Material Sciences, Civil Engineering, Electrical Engineering, Chemistry, Mathematics, Electronics Engineering and Management. All the videos have English as the medium of instruction. Hence the transcripts generated consist of words in English.

3.2 Extracting Video Content

The contents of the videos in *Videopedia* are represented by their transcripts. The transcripts are generated from the YouTube platform using the CC feature of YouTube. We also extract the metadata for each of the videos in the form of the title, description and keywords of the videos. The subtitle files available along with the videos uploaded on the Videolectures.net are used. The text available after removing the timing information from the subtitle files provide the text for the transcripts of these videos. These text files are then processed as illustrated in Section 3.4.

3.3 Extracting Webpage Content

When the browser sends a request for relevant videos, the content of the Wikipedia page is extracted at the server. For the metadata of the webpages, we consider the topic of the Wikipedia page and the summary at the top of the page. The content of the remaining part of the wikipage along with all the subsections are extracted for topic modeling. Processing according to Section 3.4 provides us the bag of words to be used as representative for the webpage.

3.4 Processing the Extracted Text

Methods	Precision	
	Before Dict Matching	After Dict Matching
VSM	0.024	0.27
PLSA	0.244	0.27
LDA	0.146	0.475
Videopedia	0.38	0.642

Table 1: Comparison of precision before and after dictionary matching.

The transcripts extracted from the videos are not 100% accurate, which necessitates correcting the transcripts by checking with a dictionary. We match the words against a dictionary provided by Wordnet and used the nearest word as a substitute for the incorrect words. Next we remove the stop-words from the remaining words in the transcripts. This leaves us with the main bag of words of the documents. The videos are educational in nature and the topics mined from the transcripts should also be educational. Hence we generate the

representative words in these transcripts by removing the non-academic words from the bag of words. This is achieved by matching them against a set of 20,000 academic words [7]. To find the best matching among the wikipages and the lecture videos, we extend the bag-of-words to include synonyms of the academic words already present in it. This will minimize the errors in case the web pages and the videos are illustrating the same topic but use synonymous words. We apply the deep learning library Word2vec [13] to extract words similar to the words in the transcript of the videos and use the newly generated vector of similar words as the representative bag-of-words for the videos. A similar approach is taken while processing the content of the Wikipedia

webpages also. The improvement in the precision of the recommendation by matching against a dictionary is shown in Table 1. We define the methods VSM and PLSA in Section 4.1.

As explained in Sections 3.2 and 3.3, we extract the metadata for both the videos and the Wikipedia pages. The metadata in both cases are also matched to the academic dictionary to remove the non-academic words. Similar to the case of processing the content of the videos and webpages, the bag-of-words for metadata are also extended by Word2vec. For each of the words in the metadata for the videos and wikipages, we find the vector of words similar to it by using Word2vec with the set of academic words as the dictionary. The union of the vectors returned by the Word2vec forms the bag of words for metadata.

3.5 Topic Modeling

After the transcripts and the bag-of-words are processed for each video, the topics of the videos are generated using LDA. LDA posits that each word in a document is generated by a topic and each document is a mixture of a finite number of topics. Each topic is represented as a multinomial distribution over words. There are a number of outputs from the LDA. We will consider $Z = \{P(z_i)\}$ – the probability distribution of the topics inside a document. These topics along with their probability distribution are then stored on the server indexed by the corresponding video-ids.

3.6 Definition of Similarity Matching

In probability theory and information theory, the Kullback-Leibler divergence (KL Div) is a measure of the difference between two probability distributions P and Q [10]. KL Div being a non-symmetric metric, we calculate the KL Div of both P from Q and Q from P and then take the average of the two. The lower the value of the KL Div between two distributions, the closer they are semantically. Hence the similarity between the videos and the webpages are calculated as the inverse of the KL Div (D_{KL}). Let $Z_{LDA}(Blog_{wiki}) = \{P(z_i)\}$, where z_i are the latent topics present in $Blog_{wiki}$ of blog pages which are returned by LDA and $Z_{LDA}(Trans_{vm}) = \{P(z_k)\}$, where z_k are the latent topics present in $Trans_{vm}$ of videos. We define the similarity measure in *Videopedia* as:

$$Sim_{Videopedia}(Blog_{wiki}, Trans_{vm}) = \left[\frac{1}{2} (D_{KL}(Z_{LDA}(Blog_{wiki}) || Z_{LDA}(Trans_{vm})) + D_{KL}(Z_{LDA}(Trans_{vm}) || Z_{LDA}(Blog_{wiki}))) \right]^{-1} \quad (1)$$

This matching between the topic models of the webpage and those of the video transcripts in the repository is also done on the server side. The recommended videos are then linked to the webpage on the browser.

3.7 Algorithm for Video Recommendation

The algorithm used to recommend the videos is given in Algorithm 1. As described in the procedure PREPAREVIDEOTOPICMODELS, our system extracts the transcripts

of the videos, removes the stop-words from these transcripts and checks the remaining words against an academic dictionary to eliminate spelling mistakes and extract the academic words from the transcripts. This facilitates the extraction of the topics related to academia. Then, the topic distribution of these videos is obtained by using LDA on these transcripts. The metadata of the videos are extracted as described in Section 3.2.

When a request is received from a website, the text of the webpage is retrieved by using the provided URL and then relevant videos are recommended as described in the procedure RECOMMENDVIDEO. The actual matching between the webpage and videos is divided into two stages. In the filtering stage, the meta data is compared in the vector space model by the cosine similarity, which removes most non relevant videos. These bag-of-words for metadata of videos are compared against the bag-of-words for metadata for the incoming website as is illustrated in the procedure MATCHMETA. The top 10% of the videos with the maximum similarity are then selected and used for matching based on topics. Let the content of the webpage W_j after checking against a dictionary be C'_j . LDA will return a set of topics $l_{C'_j}$ along with the probability distribution of the topics. Similarly, the transcript t'_i of a video v_i , has a set of topics $l_{t'_i}$ with their distribution. To find the video closest to C'_j we calculate the KL Div between $l_{C'_j}$ and $l_{t'_i}$ s for the videos v_i in the set V' , returned by the procedure MATCHMETA. Equation 1 is used to calculate the similarity and it is described in the procedure MATCHTOPICS.

4 Experimental Results

The videos used in this research paper were extracted from the repository of publicly available videos on YouTube and Videolectures.net. The webpages used as input to *Videopedia* were Wikipedia pages. In this section we provide the evaluation for our recommendation system against three baselines. The baselines are defined in the Section 4.1 and the results are presented in Section 4.2.

4.1 Baselines Used for Comparison

Topic modeling methods have been used to find latent topics from a collection of documents using probabilistic analysis of documents. However to find the semantic similarity between the documents, we need to find the similarity between the topics of these documents. There are broadly two categories of similarity measures in existing literature: (1) count similarity measures [11], (2) probabilistic similarity models [10].

In this paper, we choose one widely accepted technique from each category of models. Cosine similarity represents the count similarity measures paradigm and KL divergence represents the probabilistic similarity models. To compare the efficacy of our system, we use the VSM model [18] which uses cosine similarity and PLSA [9] which is another probabilistic topic modeling. After generating the topics through PLSA we use the KL Divergence method to find the similarity. Since in *Videopedia*, we first prune the set of videos based on metadata before applying LDA, we compare the efficiency of our algorithm with calculating the similarity between the topics generated by LDA without matching the metadata. The following subsections give the definitions necessary for the similarity measures and the topic models.

Algorithm 1 *Videopedia*: Video recommendation based on topic modeling.

Online Processing
procedure RECOMMENDVIDEO(W_j)
 $v_k = \text{PROCESSWIKIPAGE}(W_j)$
Recommend v_k for webpage W_j
end procedure

Offline Processing
procedure PREPAREVIDEOTOPICMODELS
 $V = \text{Set of videos in the Video Repository}$
for $v_i \in V$ **do**
Extract the transcript t_i of v_i
Remove stop-words from t_i and get t'_i
Check the words in t'_i against an academic dictionary to correct spelling mistakes and extract academic words
Use LDA to find the latent topics in the t'_i
Index the v_i s based on the topics
end for
end procedure

procedure PROCESSWIKIPAGE(W_j)
Extract the content of W_j as C'_j
Remove stop-words from C'_j and get C''_j
Check the words in C''_j against an academic dictionary to correct spelling mistakes and extract academic words
Select $V' = \text{MATCHMETA}(W_j, V)$
Use LDA to find the latent topics in the C'_j
for $v_i \in V'$ **do**
 t_i contains the transcript of v_i
 $r_i = \text{MATCHTOPICS}(C'_j, t'_i)$
end for
 $k = \text{argmax}_i r_i$
Return v_k for webpage W_j
end procedure

procedure MATCHMETA(W_j, V)
Extract the metadata of W_j as M_j
Remove stop-words from M_j and get M'_j
Check the words in M'_j against an academic dictionary to correct spelling mistakes and extract academic words
for $v_i \in V$ **do**
 v'_i contains the metadata of v_i
 $\text{Cosine Sim}(M'_j, v'_i) = \frac{M'_j \cdot v'_i}{\|M'_j\| \|v'_i\|}$
end for
 $V' = \text{Top 10\% of } \text{argmax}_i \text{CosineSim}(M'_j, v'_i)$
Return V'
end procedure

procedure MATCHTOPICS(C'_j, t'_i)
 $l_{t'_i} = \text{Set of latent topics in } t'_i \text{ for video } v_i$
 $l_{C'_j} = \text{Set of latent topics in } C'_j \text{ of webpage } W_j$
for each pair (l_t, l_C) in the combination of $(l_{t'_i}, l_{C'_j})$ **do**
 $\text{KLDiv}(l_C, l_t) = \frac{D_{KL}(l_C \| l_t) + D_{KL}(l_t \| l_C)}{2}$
end for
 $\text{KLDiv}(C'_j, t'_i) = \frac{\sum_{C'_j} \text{KLDiv}(l_C, l_t)}{|l_{C'_j}| |l_{t'_i}|}$
Return $[\text{KLDiv}(C'_j, t'_i)]^{-1}$
end procedure

Vector Space Model Vector Space Models represent text documents in an algebraic structure of vectors of identifiers [18]. The identifiers are often the term frequency-inverse document frequency (tf-idf) of the words in the document. Cosine similarity, which calculates the cosine of the angle between two vectors, provides a measure of the deviation between the identifiers of the document. Let the Wikipedia page be denoted by Blog_{wiki} and the transcripts for the m^{th} video, v^m , as Trans_{v^m} . In our case, while

constructing the baseline, we thus represented $Blog_{wiki}$ by its tf-idf vector $v_{wiki} = (v_{wiki_1}, v_{wiki_2}, \dots, v_{wiki_k})$. $Trans_{v^m}$ is also represented by the tf-idf vector for its transcript $v_{vid}^m = (v_{vid_1}^m, v_{vid_2}^m, \dots, v_{vid_l}^m)$, where $m = 1, \dots, n$, n represents the number of videos in the repository and l represents the number of terms in the vid^i . Thus the similarity between the Wikipedia page and the video is calculated in the VSM model by the following formula:

$$Sim_{VSM}(Blog_{wiki}, Trans_{Vid}^m) = \frac{v_{wiki} \cdot v_{vid}^m}{\|v_{wiki}\| \|v_{vid}^m\|} = \frac{\sum_{i=1}^n v_{wiki_i} \times v_{vid_i}^m}{\sqrt{\sum_{i=1}^n (v_{wiki_i})^2} \times \sqrt{\sum_{i=1}^n (v_{vid_i}^m)^2}} \quad (2)$$

PLSA Probabilistic Latent Semantic Analysis (PLSA) is a technique to statistically analyze the co-occurrence of words and documents [9]. Let $Z_{PLSA}(Blog_{wiki}) = \{P(z_i)\}$, where z_i are the latent topics present in $Blog_{wiki}$ and $Z_{PLSA}(Trans_{v^m}) = \{P(z_k)\}$, where z_k are the latent topics present in $Trans_{v^m}$. Using the probability distribution of the topics in a Wikipedia page and that in the transcripts of the videos, we find the KL Div between them and subsequently the similarity with the following formula:

$$Sim_{PLSA}(Blog_{wiki}, Trans_{v^m}) = [\frac{1}{2}(D_{KL}(Z_{PLSA}(Blog_{wiki}) \| Z_{PLSA}(Trans_{v^m})) + D_{KL}(Z_{PLSA}(Trans_{v^m}) \| Z_{PLSA}(Blog_{wiki})))^{-1} \quad (3)$$

We calculate the Sim_{PLSA} of $Blog_{wiki}$ with each of the n videos in the repository using the formula given above and the one having the highest value is recommended as the relevant video for the wikipage.

4.2 Evaluation of Recommendation

We use Precision, Recall and F-measure to evaluate the recommendation by our system. The definitions for them are provided below:

$$\text{Precision} = \frac{|RelevantVideos \cap RetrievedVideos|}{|RetrievedVideos|}$$

$$\text{Recall} = \frac{|RelevantVideos \cap RetrievedVideos|}{|RelevantVideos|}$$

$$\text{F-measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

We tested the system by recommending videos for 1,000 Wikipedia pages categorized under the same eight subjects as the e-learning videos. The distribution of the videos across the subjects is given in Fig. 2a and the distribution of the subjects among the wikipages used for testing the system are shown in Fig. 2b. We have created the ground truth of whether a video is relevant or not by checking if the wiki-category of the Wikipedia page, which is an input to *Videopedia*, matches the subject in the metadata of the relevant videos. Hence, the number of Relevant Videos for a particular webpage is often more than 1 and thus $|RelevantVideos|$ may be greater than

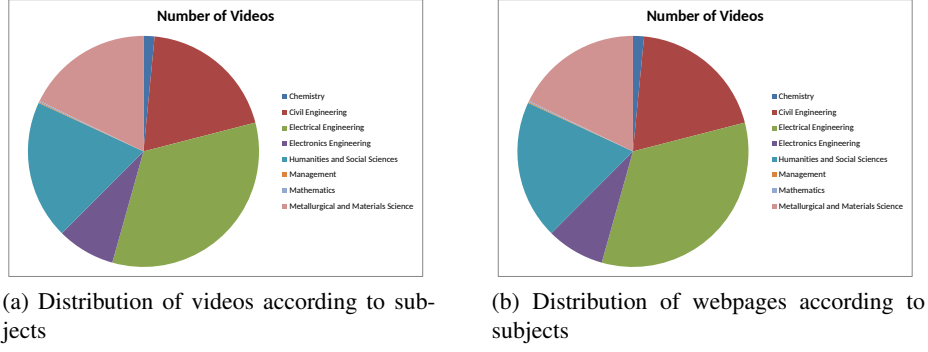


Fig. 2: Distribution of videos and webpages according to subjects

the $|RetrievedVideos|$. The Precision, Recall and F-measure values for *Videopedia* and the three baselines VSM, PLSA and LDA are presented in Table 2. We proceed to measure the effectiveness of our method by recommending more than 1 video for a webpage. The columns @ n are the results of Precision (Table 2a), Recall (Table 2b) and F-measure (Table 2c) when we recommend n videos for a webpage, where $n = 1, 5, 10$. When *Videopedia* is recommending more than one video for a particular webpage, we use the mean average precision defined below to find the precision.

$$mAP = \frac{\sum_{i=1}^n AveragePrecision_i}{n}$$

where n represents the total number of webpages for which the videos are retrieved and $AveragePrecision_i$ is the Average Precision for retrieving video for the i^{th} webpage.

Methods	Precision		
	@ 1	@ 5	@ 10
VSM	0.198	0.179	0.185
PLSA	0.0213	0.239	0.203
LDA	0.475	0.323	0.309
Videopedia	0.642	0.658	0.655

(a) Precision

Methods	Recall		
	@ 1	@ 5	@ 10
VSM	0.022	0.099	0.206
PLSA	0.021	0.133	0.226
LDA	0.053	0.180	0.343
Videopedia	0.256	0.244	0.2368

(b) Recall

Methods	F-measure		
	@ 1	@ 5	@ 10
VSM	0.040	0.128	0.195
PLSA	0.039	0.171	0.214
LDA	0.095	0.23	0.325
Videopedia	0.366	0.356	0.348

(c) F-Measure

Table 2: Videopedia compared with VSM, PLSA and LDA

As is evident from the results, *Videopedia* performs much better than previous methods, by using a refining stage based on metadata. To decide the actual percentage of videos that should be selected after matching the metadata between the incoming webpage and the videos in the repository, we calculate the the precision of the recommendations made by *Videopedia* for different percentages of videos selected. This is reported in Fig. 3b. From the precision values for each of the three recommendations, we find that *Videopedia* performs best when we select the top 10% of the videos in the repository are selected after matching the metadata. We also perform a comparative study to select

the optimal number of topics mined from the transcripts by LDA and report the findings in Fig. 3a.

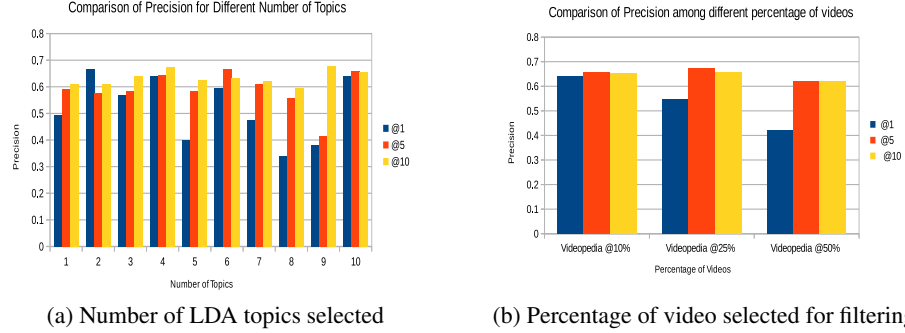


Fig. 3: Comparison of various methods

From the results reported in Figures 3b and 3a, we find that our algorithm in *Videopedia*, performs best when it selects 10% of the videos after matching the metadata of the incoming wikipedia with the metadata of the videos. The results also show that the best precision is obtained when LDA returns 10 topics from the content of the webpages and videos. This also provides a more comprehensive topic modeling of the documents.

5 Conclusions

We designed and demonstrated a novel technique in *VIDEOPEDIA* which integrates multiple media formats – namely making the educational blogs and Wikipedia pages more illustrative with multimedia contents like videos. The main contribution of this work is to automatically recommend relevant educational videos for blogs like Wikipedia. We effectively used topic modeling on automatically generated video transcripts from various video sharing platforms and using them as a representation of the video content. Our promising results show that topic modeling is a good way of video recommendations. This integrated framework reduces the users efforts in searching relevant e-learning videos. Moreover, it provides a mechanism for content-based search. We aim to scale our system for global use in academia and industry in near future. Although we focused on the educational blogs in an e-learning setup, this technique can be extended to other blogs as well.

Acknowledgments

This research has been partially supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office through the Centre of Social Media Innovations for Communities (COSMIC). This work was also partially supported by JSPS KAKENHI Grant Number 15H06829.

References

1. Cisco visual networking index: Forecast and methodology, 2013–2018. 2014.

2. J. Adcock, M. Cooper, L. Denoue, H. Pirsiavash, and L. A. Rowe. Talkminer: A lecture webcast search engine. In *Proceedings of the ACM International Conference on Multimedia*, MM '10, pages 241–250, 2010.
3. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
4. H. Chen, M. Cooper, D. Joshi, and B. Girod. Multi-modal language models for lecture video retrieval. In *Proceedings of the ACM International Conference on Multimedia*, MM '14, pages 1081–1084, 2014.
5. Z. Chen, J. Cao, Y. Song, Y. Zhang, and J. Li. Web video categorization based on wikipedia categories and content-duplicated open resources. In *Proceedings of the ACM International Conference on Multimedia*, MM'10, pages 1107–1110, 2010.
6. Z. Chen, B. Feng, H. Xie, R. Zheng, and B. Xu. Video to article hyperlinking by multiple tag property exploration. In *Proceedings of 20th International Conference on MultiMedia Modeling*, MMM'14, pages 62–73, 2014.
7. D. Gardner and M. Davies. A new academic vocabulary list. *Applied Linguistics*, 2013.
8. A. J. Head and M. B. Eisenberg. How today's college students use wikipedia for course-related research. *First Monday*, 15(3), 2010.
9. T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 50–57, 1999.
10. A. Huang. Similarity measures for text document clustering. In *Proceedings of the Sixth New Zealand Computer Science Research Student Conference*, NZCSRSC'08.
11. M. D. Lee and M. Welsh. An empirical evaluation of models of text document similarity. In *Proceedings of XXVIIth Annual Conference of Cognitive Science Society*, CogSci'05, pages 1254–1259, 2005.
12. T. Liu and J. R. Kender. Lecture videos for e-learning: Current research and challenges. In *Proceedings of the 6th IEEE International Symposium on Multimedia Software Engineering*, ISM'04, pages 574–578, 2004.
13. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.
14. E. R. Misanchuk, R. A. Schwier, and E. Boling. Visual design for instructional multimedia. In *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications*, pages 1621–1621, 1999.
15. T. Okuoka, T. Takahashi, D. Deguchi, I. Ide, and H. Murase. Labeling news topic threads with wikipedia entries. In *Proceedings of the 11th IEEE International Symposium on Multimedia*, ISM'09, pages 501–504, 2009.
16. S. Repp, S. Linckels, and C. Meinel. Question answering from lecture videos based on an automatic semantic annotation. *ACM SIGCSE Bulletin*, 40(3):17–21, 2008.
17. S. Roy, M.-T. Mak, and K. W. Wan. Wikipedia based news video topic modeling for information extraction. In *Advances in Multimedia Modeling*, pages 411–420. Springer, 2011.
18. G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., 1986.
19. C. Wang and D. M. Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 448–456, 2011.
20. Q. Zhu, M.-L. Shyu, and H. Wang. Videotopic: Content-based video recommendation using a topic model. In *Proceedings of IEEE International Symposium on Multimedia*, ISM'13, pages 219–222, 2013.