ATLAS: Automatic Temporal Segmentation and Annotation of Lecture Videos Based on Modelling Transition Time

Rajiv Ratn Shah
School of Computing, National University of Singapore, Singapore
rajiv@comp.nus.edu.sg

Yi Yu
School of Computing, National University of Singapore, Singapore
yuy@comp.nus.edu.sg

Anwar Dilawar Shaikh
Department of Computer Engineering, Delhi Technological University, India
anwardshaikh@gmail.com

Suhua Tang
Graduate School of Informatics and Engineering, UEC, Japan
shtang@uec.ac.jp

Roger Zimmermann
School of Computing, National University of Singapore, Singapore
rogerz@comp.nus.edu.sg

ABSTRACT

The number of lecture videos available is increasing rapidly, though there is still insufficient accessibility and traceability of lecture video contents. Specifically, it is very desirable to enable people to navigate and access specific slides or topics within lecture videos. To this end, this paper presents the ATLAS system for the VideoLectures.NET challenge (MediaMixer, transLectures) to automatically perform the temporal segmentation and annotation of lecture videos. ATLAS has two main novelties: (i) a SVM^{hmm} model is proposed to learn temporal transition cues and (ii) a fusion scheme is suggested to combine transition cues extracted from heterogeneous information of lecture videos. According to our initial experiments on videos provided by VideoLectures.NET, the proposed algorithm is able to segment and annotate knowledge structures based on fusing temporal transition cues and the evaluation results are very encouraging, which confirms the effectiveness of our ATLAS system.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; K.3.1 [**Computers and Education**]: Computer Uses in Education—*Distance learning*

Keywords

Lecture video segmentation; lecture video annotation; lecture video recommendation

1. MOTIVATION AND BACKGROUND

The ATLAS system presented in this paper is our solution to the ACM Multimedia 2014 Grand Challenge on automatic temporal segmentation and annotation. ATLAS stands for *automatic temporal segmentation and annotation of lecture videos based on modelling transition time*. The number of lecture video recordings on the web has increased rapidly due to the ubiquitous availability of cameras and the affordable network infrastructure. However, the accessibility and traceability of lecture video content is still a challenging task. To solve this problem, ATLAS first predicts temporal transitions (TT_1) using supervised learning on video content and temporal transitions (TT_2) by text (transcripts and/or slides) analysis using an N-gram based language model. In the next step, TT_1 and TT_2 are fused by our algorithm to obtain a list of transition times for lecture videos. Moreover, text annotations corresponding to these temporal segments are determined by assigning the most frequent N-gram token of the subtitle resource tracks (SRT) block under consideration (and similar to the N-gram token of slide titles, if available). In this way, the proposed ATLAS system improves the automatic temporal segmentation and annotation of lecture videos so that online learning becomes much easier and users can search sections within a lecture video.

ATLAS introduces new algorithms to automatically segment lecture videos based on video and text analysis. Furthermore, it automatically annotates the segments using an N-gram based language model. A color histogram of a keyframe at each shot-boundary is used as a visual feature to represent the slide transition in the video content. The relationship between the visual features and the transition time of a slide is established with a training dataset of lecture videos provided by the grand challenge organizers, using a machine-learning SVM^{hmm} technique. The SVM^{hmm} model predicts temporal transitions for a lecture video. Furthermore, our solution can help in recommending similar content to the users using text annotations as keywords for searching. The ATLAS system can determine temporal segments and annotations of lecture videos offline rather than at search time, therefore, our system is time-efficient and scales well to large repositories of lecture videos. Our initial experiments have confirmed that our system recommends reasonable temporal segmentations and annotations for lecture videos.

MM '14, November 3–7, 2014, Orlando, Florida, USA.

http://dx.doi.org/10.1145/2647868.2656407.

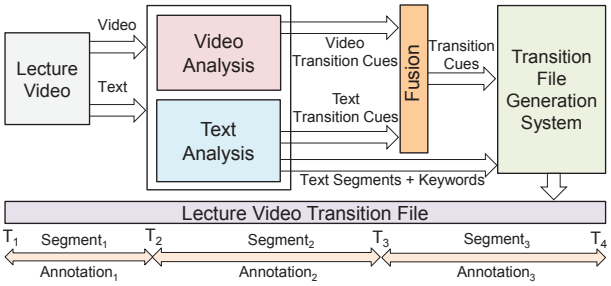

Figure 1: System Framework of ATLAS.

The paper is organized as follows. In Section 2, we review related work and Section 3 describes the ATLAS system. The evaluation results are presented in Section 4. Finally, we conclude the paper with a summary in Section 5.

2. RELATED WORK

Automatic temporal segmentation and annotation of lecture videos is a challenging task, since it depends on many factors such as speaker presentation style, characteristic of camera (*i.e.*, video quality, static or dynamic position/view, *etc*), and others. Moreover, it is a cross-disciplinary area which requires knowledge of text analysis, visual analysis, speech analysis, machine learning and others. In the last a few years, several researchers attempted to solve the problem of automatic segmentation and annotation of lecture videos. Earlier work [5, 7–9] attempted to segment videos automatically by exploiting visual, audio and linguistic features. Fan *et al.* [4] tried to match slides with presentation videos by exploiting visual content features. Chen *et al.* [3] attempted to automatically synchronize presentation slides with the speaker video. Repp *et al.* [10] proposed the segmentation and annotation of audiovisual recordings based on automated speech recognition. Recently, Bhatt *et al.* [1] and Che *et al.* [2] attempted to automatically determine the temporal segmentation and annotation for lecture videos.

3. SYSTEM OVERVIEW

Our system has several novel components which together form its innovative contributions (see Figure 1 for the system framework). The ATLAS system performs the temporal segmentation and annotation of a lecture video in three steps. First, transition cues are predicted from the visual content, using supervised learning described in Section 3.1. Second, transition cues are computed from the available texts using an N-gram based language model described in Section 3.2. Finally, transition cues derived from the previous steps are fused to compute the final temporal transitions and annotations with text, as described in Section 3.3.

3.1 Prediction of Video Transition Cues

A lecture video is composed of several shots combined with cuts and gradual transitions. Kucuktunc *et al.* [7] proposed a video segmentation approach based on fuzzy color histograms which detects shot-boundaries. Therefore, we train two machine learning models using a SVM^{hmm} [6] technique by exploiting the color histograms (64-D) of keyframes to detect the slide transitions automatically in a lecture video. As described in the later Section 4.1, we use lecture videos (V_T) with known transition times as the test set and the remaining in the dataset as the training set. We

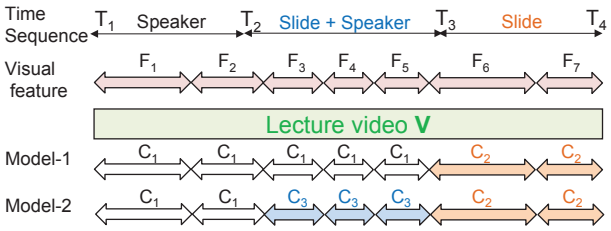

Figure 2: Slide transition models.

employ human annotators to annotate ground truths for lecture videos in the training set (see Figure 2 for an illustration of the annotation with both models).

First, a SVM^{hmm} model M_1 is trained with two classes C_1 and C_2. Class C_2 represents the segment of a lecture video when only a slideshow is visible (or the slideshow covers a major fraction of a frame) and class C_1 represents the remaining part of the video (see Model-1 in Figure 2). Therefore, whenever a transition occurs from a sequence of classes C_1 (*i.e.*, from speaker only or, both speaker and slide) to C_2 (*i.e.*, slideshow only), it indicates a temporal transition with high probability in the majority of cases. However, we find that for some videos this model detects very few transitions (less than five transitions only). We notice that there are mainly three reasons for this issue, first, when lecture videos are recorded with a single shot, second, when the transition occurs from a speaker to a slideshow but the speaker is still visible in the frame most of the time, and third, when the transition occurs between two slides only.

To resolve the above issues, we train another SVM^{hmm} model M_2 by adding another class C_3, which represents the part of a video when a slideshow and a speaker are both visible. We use this model to predict transitions from only those videos for which M_1 predicted very few transitions. We do not use this model for all videos due to two reasons. First, the classification accuracy of M_1 is better than that of M_2 when there is a clear transition from C_1 to C_2. Second, we want to focus on only those videos which exhibit most of their transitions from C_1 to C_3 throughout the video (this is the reason M_1 was predicting very few transitions), hence, a transition from a sequence of classes C_1 to C_3 is considered a slide transition for such kind of videos.

3.2 Computation of Text Transition Cues

3.2.1 *Preparation*

In the preparation step, we convert slides (a PDF file) of a lecture video to a HTML file using Adobe Acrobat software. However this can be done with any other proprietary or open source software as well. The benefit of converting the PDF to an HTML file is that we obtain the text from slides along with their positions and font sizes, which are very important cues to determine the title of slides.

3.2.2 *Title/Sub-Title Text Extraction*

Algorithm 1 extracts titles/sub-titles from the HTML file derived from slides, which represent most of the slide titles of lecture videos accurately. A small variation of this algorithm produces the text content of a slide by extracting the text between two consecutive title texts.

3.2.3 *Transition Time Recommendation from SRT File*

We employ an N-gram based language model to calculate the relevance score R for every block of thirty tokens from

Algorithm 1 Title/sub-title text extraction from slides

1: **procedure** TitleOfSlides(S)
2: INPUT: A HTML file for slides (H)
3: OUTPUT: A list of title text T
4: extractFontFreq(H, $fontList$, $freq$) ▷ this function finds all font and their frequency counts in slides.
5: $titleFontSize$ = findTitleFontSize($fontList$, $freq$) ▷ this function determines the font size of the title of slides.
6: $numSlides$ = findNumSlides($titleFontSize$) ▷ this function calculates the approx number of slides.
7: T = findTitleText($titleFontSize$, $position$) ▷ this function determines the text for titles of all slides which located in top 1/3 of vertically or 2/3 of horizontally in slides.
8: **end procedure**

a SRT file. We use a hash map to keep track of all N-gram tokens and their respective term frequencies (TF). The relevance score is defined by the following equation,

$$R(B_i) = \sum_{j=1}^{N} \sum_{k=1}^{n} W_j * w(t_k),$$

$$\text{and } w(t_k) = \frac{TF(t_k|B_i) * log(TF(t_k|SRT) + 1)}{log \frac{TF(t_k|SRT)+1}{TF(t_k|B_i)}},$$

where, $TF(t_k|B_i)$ is the TF of an N-gram token t_k in a block B_i and $TF(t_k|SRT)$ is the TF of the token t_k in the SRT file. N is the N-gram count (we consider up to $N = 3$, *i.e.*, trigram), W_j is the weight for different N-gram counts such that the sum of all W_j is equal to one, and n is the number of unique tokens in the block B_i. We place more importance to a higher order N-gram count by assigning high values to W_j in the relevance score equation.

If slides of a lecture video are available then we calculate the approximate number of slides ($numSlides$) using the Algorithm 1. We consider the $numSlides$ number of SRT blocks with the highest relevance scores to determine transitions using text analysis. We infer the start time of these blocks from the hash-map and designate them as the temporal transitions derived from the available texts.

3.3 Transition File Generation

We fuse the temporal transitions derived from the visual contents and the speech transcript file by replacing two transitions less than ten seconds apart by their average transitions time and keeping the remaining transitions as the final temporal transitions for the lecture video. Next, we compare N-gram tokens of blocks corresponding to the final temporal transitions and calculate their similarity with N-gram tokens derived from the title of slides. We assign the most similar N-gram token of a block B_i as a text annotation A for a temporal segment which consists of B_i. If slides of lecture videos are not available then an N-gram token with high TF is assigned as a text annotation for the lecture segment.

4. EVALUATION

4.1 Dataset and Experimental Settings

A total of 65 videos (V) were provided with several metadata annotations such as speech transcriptions, slides, transitions, *etc.*, for the VideoLectures.NET Challenge. A transition file contains details of all transitions and corresponding title texts for a lecture video. Therefore, details in the transition file are treated as ground truth for the lecture video segmentation and annotation task. V_T is the test set for the evaluation of our approach, consisting of videos with transition files.

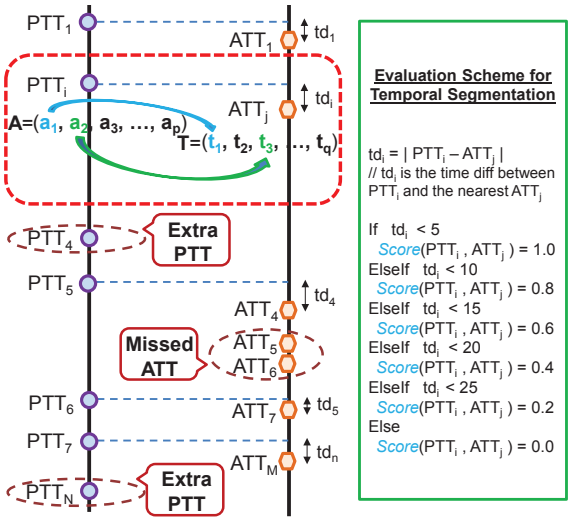

Figure 3: Mapping of PTT, ATT and their respective text to calculate precision, recall and F-1 scores.

4.2 Results

The ATLAS system determines the temporal transitions and the corresponding annotations of lecture videos, with details described earlier in Sections 3.1, 3.2 and 3.3. To evaluate the effectiveness of our approach, we compute precision, recall and F-1 scores for each video in V_T. For a few videos in V_T, precision, recall and F-measure values are very low because our SVM^{hmm} models are not able to detect transitions in lecture videos if lectures are recorded with a single shot, or without zoom-in, zoom-out, or when the slide transitions occur between two slides without any other change in the background. For example, precision and recall for the lecture video *cd07_eco_thu* are *zero*, since only a speaker is visible in the whole video except for a few seconds at the end when both the speaker and a slide consisting of an image with similar color as the background are visible. Therefore, for videos in which our machine learning techniques are not able to detect transitions, we determine transitions from analyzing the speech transcripts (and the text from slides if available) using an N-gram based language model as described in the earlier Section 3.2.

For an evaluation of the temporal segmentation, we connect one predicted transition time (PTT) with only one nearest actual transition time (ATT) from the provided transition files. It is possible that some PTTs are not connected with any ATT and vice versa, as shown in Figure 3. For example, PTT_4 and PTT_N are not connected with any ATT. Similarly, ATT_5 and ATT_6 are not connected with any PTT. We refer to these PTTs and ATTs as $ExtraPTT$ and $MissedATT$, respectively. We compute the score for each (PTT_i, ATT_j) pair based on the time difference between them, by employing a relaxed approach as depicted in Figure 3 because it is very difficult to predict exactly the same transition time at the granularity of seconds. Therefore, to evaluate the accuracy of the temporal segmentation,

Table 1: Evaluation of temporal segmentation for the lecture videos in test set V_T.

	Segmentation Accuracy with Visual Transition Cues (I)			Segmentation Accuracy with Text Transition Cues (II)			Segmentation Accuracy with Fused Transition Cues (III)		
Video Name	**Precision**	**Recall**	**F-1**	**Precision**	**Recall**	**F-1**	**Precision**	**Recall**	**F-1**
sparsemethods_01	0.536	0.728	0.618	0.245	0.185	0.211	0.393	0.638	0.486
scholkopf_kernel_01	0.434	0.451	0.442	0.186	0.255	0.219	0.258	0.506	0.341
denberghe_convex_01	0.573	0.487	0.526	0.397	0.296	0.339	0.452	0.496	0.473
bartok_games	0.356	0.246	0.291	0.156	0.938	0.268	0.169	0.831	0.281
abernethy_learning	0.511	0.192	0.279	0.340	0.625	0.441	0.379	0.600	0.465
agarwal_fgc	0.478	0.287	0.358	0.440	0.367	0.400	0.358	0.393	0.375
abernethy_strategy	0.600	0.235	0.338	0.518	0.496	0.507	0.500	0.435	0.465
cd07_eco_thu	0	0	-	0.166	0.154	0.160	0.183	0.154	0.167
szathmary_eol	0.545	0.988	0.702	0.109	0.225	0.147	0.307	0.825	0.447
2011_agarwal_model	0.350	0.088	0.140	0.366	0.331	0.348	0.366	0.331	0.348
2010_agarwal_itl	0.571	0.174	0.267	0.371	0.339	0.354	0.320	0.348	0.333
leskovec_mlg_01	0.492	0.451	0.471	0.356	0.251	0.294	0.397	0.419	0.408
taylor_kmsvm_01	0.650	0.325	0.433	0.260	0.232	0.245	0.391	0.489	0.435
green_bayesian_01	0.473	0.492	0.483	0.362	0.353	0.357	0.339	0.539	0.416
icml08_agarwal_mpg	0.200	0.012	0.023	0.363	0.352	0.357	0.500	0.121	0.190
nonparametrics_01	0.384	0.571	0.459	0.231	0.331	0.272	0.301	0.584	0.397
bubeck_games	0.655	0.465	0.543	0.280	0.452	0.347	0.379	0.574	0.456
Overall_score	**0.459**	**0.364**	**0.375**	**0.303**	**0.363**	**0.310**	**0.352**	**0.487**	**0.381**

we use the following equations to compute precision and recall, and then compute F-measure (F-1 score) using the standard formula $(2 \times \text{precision} \times \text{recall})/(\text{precision} + \text{recall})$.

$$Precision\_segmentation = \frac{\sum_{k=1}^{r} Score(PTT_i, ATT_j)}{N}$$

$$Recall\_segmentation = \frac{\sum_{k=1}^{r} Score(PTT_i, ATT_j)}{M}$$

where N is the number of ATTs, M is the number of PTTs and r is the number of (PTT_i, ATT_j) pairs.

Table 1 shows the precision, recall and F-1 scores for the temporal segmentation of the lecture videos, (I) when visual transition cues are predicted by our SVM^{hmm} models, (II) when text transition cues are predicted by our N-gram based approach, and (III) when the visual transition cues are fused with text transition cues. Furthermore, it shows that the proposed scheme (III), improves the average *recall* much and the average *F-1* slightly, compared with the other two schemes. Therefore, the transition cues determined from the text analysis are also very helpful, especially when the supervised learning fails to detect temporal transitions.

5. CONCLUSIONS

The proposed ATLAS system provides a novel and time-efficient way to automatically determine the temporal segmentation and annotation of lecture videos. First, it determines the temporal segmentation by fusing the transitions cues computed from the visual contents and the text analysis. Second, it annotates the texts corresponding to the determined temporal transitions. ATLAS can effectively segment and annotate lecture videos to facilitate the accessibility and traceability within their content. In our future work, we plan to extend the approach to apply supervised learning on all available texts using features derived from a confusion matrix based on an N-gram language model.

ACKNOWLEDGMENTS

This research has been 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).

6. REFERENCES

[1] C. A. Bhatt, A. Popescu-Belis, M. Habibi, S. Ingram, S. Masneri, F. McInnes, N. Pappas, and O. Schreer. Multi-factor Segmentation for Topic Visualization and Recommendation: the MUST-VIS System. In *ACM Multimedia*, pages 365–368, 2013.

[2] X. Che, H. Yang, and C. Meinel. Lecture Video Segmentation by Automatically Analyzing the Synchronized Slides. In *ACM Multimedia*, pages 345–348, 2013.

[3] Y. Chen and W. J. Heng. Automatic Synchronization of Speech Transcript and Slides in Presentation. In *IEEE ISCS*, volume 2, pages 568–571, 2003.

[4] Q. Fan, K. Barnard, A. Amir, A. Efrat, and M. Lin. Matching Slides to Presentation Videos using SIFT and Scene Background Matching. In *ACM Multimedia*, pages 239–248, 2006.

[5] A. Haubold and J. R. Kender. Augmented Segmentation and Visualization for Presentation Videos. In *ACM Multimedia*, pages 51–60, 2005.

[6] T. Joachims, T. Finley, and C.-N. Yu. Cutting-plane Training of Structural SVMs. *In Machine Learning Journal*, 77(1):27–59, 2009.

[7] O. Kucuktunc, U. Gudukbay, and O. Ulusoy. Fuzzy Color Histogram-based Video Segmentation. *In Computer Vision and Image Understanding*, 114(1):125–134, 2010.

[8] M. Lin, J. F. Nunamaker Jr, M. Chau, and H. Chen. Segmentation of Lecture Videos based on Text: A Method Combining Multiple Linguistic Features. In *IEEE Hawaii ICSS*, pages 9–17, 2004.

[9] C.-W. Ngo, T.-C. Pong, and T. S. Huang. Detection of Slide Transition for Topic Indexing. In *IEEE ICME*, volume 2, pages 533–536, 2002.

[10] S. Repp, J. Waitelonis, H. Sack, and C. Meinel. Segmentation and Annotation of Audiovisual Recordings based on Automated Speech Recognition. In *Springer IDEAL*, pages 620–629. 2007.