Published as: Che, X., Yang, H., & Meinel, C. (2013, September). Tree-structure outline generation for lecture videos with synchronized slides. In e-Learning and e-Technologies in Education (ICEEE), 2013 Second International Conference on (pp. 87-92). IEEE.

Tree-Structure Outline Generation for Lecture Videos with Synchronized Slides

Xiaoyin Che
Hasso Plattner Institute
Prof.-Dr.-Helmert-Str. 2-3
14482 Potsdam, Germany
xiaoyin.che@hpi.uni-potsdam.de

Haojin Yang
Hasso Plattner Institute
Prof.-Dr.-Helmert-Str. 2-3
14482 Potsdam, Germany
haojin.yang@hpi.uni-potsdam.de

Christoph Meinel
Hasso Plattner Institute
Prof.-Dr.-Helmert-Str. 2-3
14482 Potsdam, Germany
christoph.meinel@hpi.uni-potsdam.de

***Abstract*—In this paper we propose a solution to generate tree-structure outline for lecture videos by analyzing their synchronized slides, by which detailed lecture overview can be automatically provided to E-learning portal users. Starting with OCR (*Optical Character Recognition*) result, we reconstruct the content of each slide. After that, we explore logical relations between slides, in order to make them hierarchical. And all potential redundant content will also be removed. Our evaluation shows that, based on our test dataset, the final outline achieved retains about 1/4 of the original texts from all slides and is organized well in an up-to-6-level tree structure. Furthermore, the average accuracy of all slide titles, which are undoubtedly the most important information, reaches 86%.**

Keywords—Tree-Structure Outline, Slides Analysis, OCR.

I. Introduction

E-learning nowadays is quite widespread all over the world and video is one kind of fundamental material in most E-learning systems. However, people always find it difficult to judge whether a lecture video is exactly what they need by just glancing at the video title. More detailed but still accurate information should be provided to the E-Learning portal users.

Over years the most common method to do so is tagging, which has been researched in various ways. But no matter the tags come from user feedback[1, 2], or derive from automatically semantic analysis[1, 3, 4], the only problem tagging can solve is "What is it", not "How is it". Perhaps it is enough for the curious audience but not for the purposeful learners. However, some investigations reveal that lecture outlines help the students who taking online courses a lot[5, 6]. However, the realistic problem is how to achieve the lecture outline and obviously, this work should not be done manually due to the huge potential consumption of time or/and money.

Since more and more lecturers use slides instead of blackboard when giving their lectures, extracting the content of the slides can be an excellent choice to generate outline, because generally, slides are exactly the outline of the lecture, detailed and accurate. Some attempts have already been made in generating outline from lecture slides[7], but we want to make the outline better organized and more friendly to the users, rather than simply listing the content of the slides.

In this paper, we propose a solution to do this task based on OCR result. OCR technology enables us to get the textual data from the slides automatically, with a fairly good accuracy, approximately 92% about the characters and 85% about the words[8]. Then we can reconstruct each slide into an up-to-3-level content tree by analyzing the size, location and all other possible attributes of the texts achieved by OCR process (*all attributes saved in data structure 'text-line' together with the text*). Next we will remove repeated or useless slides and explore any logical relations between the remainings, by which the slides can be set hierarchical. Along with the up-to-3-level intra-slide content tree, the complete outline generated by our solution will be stored in an up-to-6-level tree structure.

The rest of this paper will be organized as follow: Section II shows the framework of our solution while section III, IV and V illustrate the details. And evaluation and conclusion can be found in Section VI and Section VII respectively.

II. Solution Framework

Figure 1 illustrates the diagram of proposed solution, including three main parts, raw resources pre-processing, intra-slide content reconstruction and inter-slides analysis. Generally there are two main challenges in our research, to analyze the slide layout or logic and the robustness on OCR accuracy problem. We have to take both of them under consideration in all steps.

III. Raw Resources Pre-Processing

A. Removing Potential Logo

Some lecture or presentation slides, especially those built on a university or company template, always have a logo or badge. When existing, logo appears in the same position of almost every slide, commonly in a corner. Due to the large size and probable upper location, the logo has a great chance to be recognized as a major part of a slide by our following steps, such as title, which may drastically damage the real content structure.

To solve this problem, we employ a location-based checking scheme, in which the potential logo will be singled out and removed. Firstly, all the text-lines recognized in the first slide will be set as logo-candidate. Then the checking process starts from the second slide: if a text-line is in the same position of a logo-candidate and also has same text content, it will be counted, or else, be added as a new logo candidate. After checking all the text-lines possible, the logo candidates with a low counting number will be ignored, and this number is about

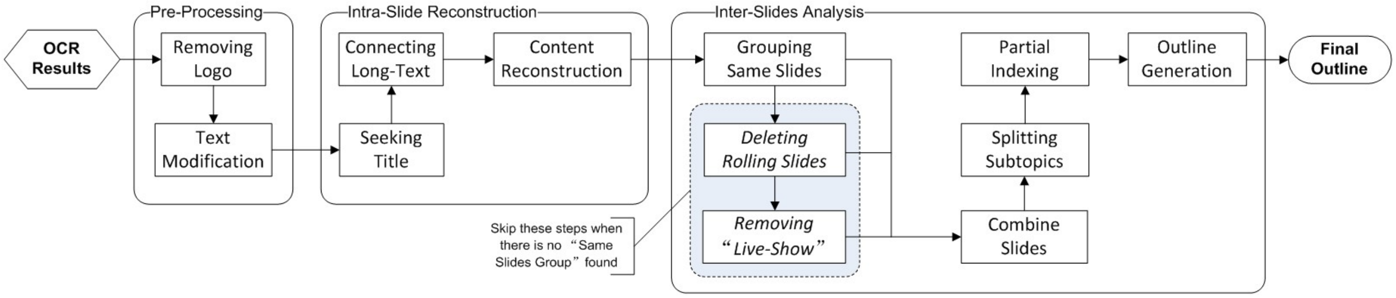


Fig. 1. Diagram of proposed solution framework

1/8 of the total number of slides. Finally all the text-lines, which are exactly matching or being absolutely included in the location of any remaining logo candidate, will be removed from the slides permanently.

In order to avoid removing some non-logo but logo-like texts, for example a same title shared by multiple slides in the same location, this checking scheme will only be applied in the edges of the slides. Despite the logo, the footline or some other template-based texts will also be removed in this step.

B. Ill-Recognized Text Modification

There is no reason to offer the meaningless 'weird' string to the users, in other words, ill-recognized text-lines must be fixed. In our research, text-lines will be checked by splitting into words. If the average word length is shorter than 2 characters, this text-line will be discarded entirely.

Otherwise, a text-line can also be shortened by eliminating ill-recognized words, which include continuous short words, a word with an abnormally long length or containing too much symbols. Besides, a special dictionary for frequently used short words or professional initials such as 'a', 'is' or 'CS' is used to keep these meaningful short words from being deleted.

IV. INTRA-SLIDE RECONSTRUCTION

A. Seeking Title

Title is the most important content for a slide. Generally, a title has 3 features: bold, locating in the upper part of the slide and being separated with other texts. And in our research, potential subtitle will be considered as a part of the title in order to avoid affecting the process of following steps. Furthermore, if the title is long, it may occupy multiple rows. So we will search for up to 3 text-lines as title candidates. After all the potential title candidates found, they will be sorted by the location logic (*top-down and left-right*) and combined together as the title.

When seeking a title candidate, all the requirements below should be considered (*with slide resolution 1024×768*):

1) Higher than the average or 30 pixels.
2) Vertically locates in the top 1/3 of the slide.
3) Horizontally locates in the left 2/3 of the slide.
4) Not closer than 10 pixels to any border.
5) If it is not the only text-line matching requirements above, not far from the previous one, neither vertically nor horizontally.

B. Connecting Continuous Text-lines

When people use long statements, descriptions or definitions in the slides, these long texts always occupy multiple rows. It is natural for human to read them continuously, but not for OCR process, by which several independent text-lines will be achieved instead. Besides, sometimes OCR may separate the text in the same row when there are a few words or symbols in the middle cannot be recognized. Thus, one continuous-text connecting process is strongly needed to solve these problems.

Here we use t_i, t_{i-1}, t_{i+1} to represent the current, previous and next text-line. If t_i and t_{i-1} locate in the same row, the only requirement for a potential combination is that the gap between them is not too large. And if t_i and t_{i-1} locate in different rows, the requirements can be described below:

1) t_i starts with a lower case letter and t_{i-1} starts with a capital letter or a number.
2) The vertical distance between t_i and t_{i-1} is not larger than the vertical distance between t_{i+1} and t_i.
3) Horizontally the left-edge of t_i should be near the left-edge of t_{i-1}, and the right-edge of t_i should not be much beyond the right-edge of t_{i-1}.
4) The width of t_{i-1} should at least reach the average width of (*if possible*) t_{i-2}, t_{i-1}, t_i, t_{i+1}, t_{i+2}.

After each one combination, the right text-line (*when in same row*) or the bottom text-line (*when in different rows*) will be removed and its text will be set at the end of the remaining text-line as extension. Meanwhile, the parameters of the remaining text-line, such as height, width and location info should be updated, by which it can be immediately considered in a new potential combination attempt, when a long text occupies more than 2 rows.

For those ambiguous conditions in this step, we prone to not do the combination, because incorrectly combining two independent texts can be more harmful to the content structure of the slide than separating a continuous text into two.

C. Content Structure Reconstruction

In this step, the text-lines within one slide will be organized in hierarchy, up to 3 levels. Generally, which level a text-line belongs to mainly depends on its location inside the slide, and its height is also an influential factor. In our research, up to 3 horizontal coordinates (*derived from the left-edge of selected text-lines*) will be found out to mark the levels. The text-lines belong to those 3 potential levels consist the text system of this slide, but there are always some out-of-system text-lines, such as a page number or a figure description, interfering our effort to locate the text-system.

It is natural to begin with searching for the first level text-lines, but since all those in-system text-lines generally gather together by their left-edges, singling out one of them accurately is enough to help us find the others. Furthermore, because of the inherent OCR problem, for example the height of the words 'glory' 'name' or 'PAPER' can be very different even with same font and size, a first level text-line may not qualify by our restriction on height, which reassures our intention to just find a definite in-system text-line first, rather than a in-system first level text-line.

According to the custom that most lecturers write their slides content from the left horizontally and from the top vertically, we will firstly search the text-line with their left-edge locating in the left-top quarter of the slide, in order to find the leftmost text-line with its height higher than the average. If there is no suitable text-line found (*in some cases a large picture will occupy the left-top quarter of the slide*), we will further search in a larger area, 3/4 of both left and top.

As mentioned before, this selected text-line is only in-system, not surely in first level, so we have to figure out whether there is a more left-located in-system text-line of the current selected one. Using x to represent the coordinate of the left-edge of current selected text-line, we will find every text-line t_i with its left-edge locates in $[x-5, x+5]$ and test its previous text-line t_{i-1} on whether its left-edge locates in $[x-75, x-15]$. If so, that t_{i-1} will be the new selected one and x will be updated, then this process will be repeated. The final x will be the first level coordinate. Next, we will try to find the horizontal coordinates for potential second and third level, if the first level exists, by finding a text-line with its left-edge locating in $[x+15, x+75]$ to get y, and then in $[y+15, y+125]$ to get z.

Now, we can set the hierarchy for all possible text-lines. Any t_i starting from $[x-5, x+5]$ will be set to level 1, and a looser measurement calculate by the gap between x and y, $m_l = (y-x)/2$, will be applied either. Any t_i starting from $[x-m_l, x+m_l]$ while t_{i-1} has already been confirmed to be in-system, will also be set to level 1. The decision process for level 2 or 3 is absolutely the same. Those out-of-system text-lines which cannot fit any level range will be set as level 0.

In addition, we employ an upgrading scheme when a text-line will be set as a level not close to its previous text-line's level, in order to avoid any logic problem in the content structure. Finally we will calculate the rate of how many text-lines have been included in the hierarchy system, by which a slide will be marked as 'well-organized' or 'ill-organized' as a slide attribute. This attribute may be useful in further steps.

V. Inter-Slides Analysis

A. Grouping Same Slides

The synchronized slides derived from the real-time lecture are always different from the original slides, not only due to the accuracy problem of OCR process, but also because many lecturers rarely run their presentations exactly following the slides prepared. It is natural for a lecturer to roll back to previous slide when he or she tries to do further explanation. And switching off the slides to a projector or a website happens also frequently. Both these situations will result in unexpected slides repeating, which we need to find out.

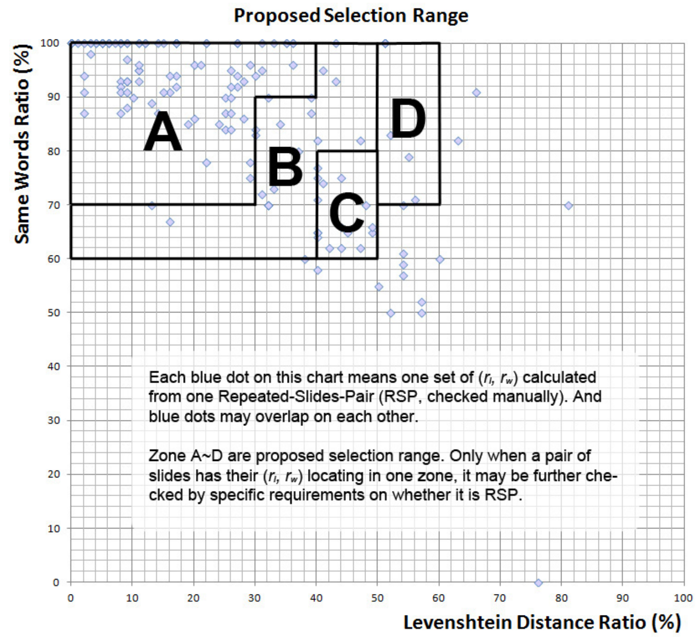


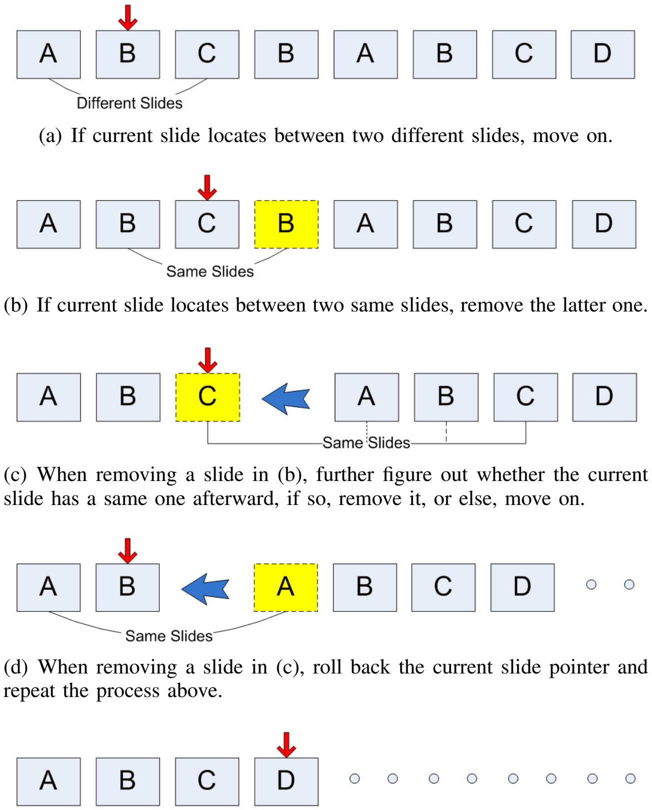
Fig. 2. Statistics of (r_l, r_w) and proposed selection range

In our research we use Levenshtein Distance[9] to help comparing slides. The Levenshtein Distance between two strings can be simply explained as the minimum number of single-character edits (*including insertion, deletion and substitution*) required to change one string into another. We calculate the ratio r_l of the Levenshtein Distance to the length of the longer string, and take r_l as one of the two main measurements. The other is the ratio of the shared words number to the total words number, which is marked as r_w.

When evaluating the similarity, all the texts in one slide, including title and text-lines, will be connected together as a long string. Then apparently, if these two long strings are similar, r_l should be small while r_w should be large. Based on this fundamental fact and after a lot of experiments, we propose the decision strategy for this problem illustrated in Figure 2 and further explained below:

Zone A: Directly accepted as RSP
Zone B: When the difference rate between two titles $r_t < 0.25$ and non-title texts longer than 50 characters, accepted as RSP
Zone C: When $r_t < 0.2$ and non-title texts longer than 100 characters, accepted as RSP
Zone D: When $r_t = 0$ and non-title texts longer than 200 characters, accepted as RSP

We won't try to expand the selection zone because the damage of falsely accepting a RSP can be severer than missing one. After going through all pairs of slides, RSPs will be saved in groups rather than pairs, because it is very likely for a slide to repeat more than twice.



(a) If current slide locates between two different slides, move on.

(b) If current slide locates between two same slides, remove the latter one.

(c) When removing a slide in (b), further figure out whether the current slide has a same one afterward, if so, remove it, or else, move on.

(d) When removing a slide in (c), roll back the current slide pointer and repeat the process above.

(e) Finally get an unrepeated slides sequence.

Fig. 3. An example of removing ‘rolling’ slides

B. Deleting Rolling Slides

As mentioned in Chapter V-A, the natural lecturer’s behavior of ‘rolling’ slides will result in the unexpected repeating, which should be fixed. In this step, we will try to remove those repeated slides according to the common ‘rolling’ operation. Figure 3 shows an example.

When removing a slide, we will first check whether there is something worth retaining, such as a text-line better recognized than those text-lines in same position of other slides within a same slide group, rather than deleting it directly. If so, the better recognized text-line will overwrite the relevant in the retained slide or been inserted in the appropriate location when there is no relevant in the retained slide. In this case, the location will be decided by the contexts. Due to the high similarity of the slides in the group, it is easy to find some text-lines as matching contexts.

C. Removing Live Show

Sometimes in a lecture, the lecturer may switch off the slides to show some websites on the screen or show some products by a projector, which is called ‘Live Show’ in our research. It helps to improve the vividness of the lecture but can be another trouble for us. In our case, the content of the ‘Live Show’ is meaningless and should be removed.

It is fortunate that generally when the lecturer switches the slides back on the screen at the end of the ‘Live Show’, the same slide when switched off will reappear. And this switch-edge slide will be recognized as two independent slides with similar contents, and then can be gathered into the same slide groups. Meanwhile, the manifestation of ‘Live Show’ in synchronized slides is just like lots of key frames extracted from a natural video, which result in a large group of ill-recognized slides (*refered in Chapter IV-C*) having apparently short display time.

Based on these facts, we will check the slides locating between neighboring same slide pairs. If over half of them are ill-recognized and the average display time is less than 5 seconds, this sequence will be marked as ‘Live Show’ and removed totally.

D. Combining Continuous Slides

If a lecturer wants to talk more about one important topic, its content may cover several continuous slides with a same title with or without potential numbers. Since these slides should be considered as a whole in the content structure, combining them together correctly is necessary.

If the titles of two continuous slides are same, or only differ of 1 or 2 characters, their content should be combined. And if these titles have typical signs in them, such as ‘(1/3)’, ‘IV’ or‘<a>’, we allow one whole word difference between the rest part of two titles.

When combining, the longer title will be retained and the content of the second slide will be added at the end of the first slide in general. But if the second slide is used to further explain one subtopic of the first slide, in other words if the first level text-line of the second slide matches a same or lower level text-line of the first slide, the relevant contents will be inserted into the corresponding position.

E. Searching Subtopic Borders

Generally a lecture is always consisted by several subtopics, but only some of them have apparent signs to indicate. When existing, these slides with signs, just like borders, certainly appear in multiple times and easily split all slides into several segments which may greatly help us to organize them into tree-structure. In our research, 3 kinds of slide will be identified as subtopic border: tag-page, split-page and section-page. Figure 4 shows examples of these 3 kinds of special slides.

1) Tag-Page: A tag-page in fact is an outline of the whole slides, with a special title such as ‘Agenda’, ‘Topics’ or ‘Outline’. For each appearance, one certain text-line will be highlighted to indicate that this subtopic will be discussed in the following slides until the next appearance of the tag-page. Obviously, the tag-page will be recognized as several independent and discontinuous slides with similar content, and further included into the same slide groups. After processing, the subtopic in tag-page will be in level 1 while the following slides in level 2.

2) Split-Page: Generally the only content in a split-page is a prompt of the following slides, and for most cases, locates in the center of the slide rather than the title position. Different from tag-page, split-pages are completely independent slides but have same format, which makes them easy to be found. After processing, split-pages will be in level 1 while others between them in level 2.

(a) Tag-Page (b) Split-Page (c) Section-Page

Fig. 4. Examples of Subtopic Border Slides

3) Section-Page: Not like tag-page or split-page above, a section-page has all the features a common slide may have, expressing definitions, explaining algorithms or showing pictures. The only deference is in the title, which contains some special border words such as 'Part 1:' 'Theme two' or 'Topic III' e.g. After processing, a virtual slide with the rest part of the special title as content will be created and set to level 1, while other slides between virtual slides will be in level 2.

F. Searching Partial Indexes

In some cases, a slide can be the partial index of several following slides, which also provides some clues for us. These index-pages are quite easy to be recognized because their text-lines are always the same or highly similar to the titles of following slides. An index-page can be in level 1 or level 2, and obviously its following slides will be in one level lower.

Sometimes after all the effort above, most of the slides keep independent. In this case, we will try to gather them by their own titles. Simply speaking, if some continuous slides, or most of the slides in a small interval, share some words (*nouns only*), they will be considered as a whole and concluded under a virtual index-page, which will be created at the beginning of this slide sequence. The word difference caused by plural will be recognized and be further ignored.

G. Final Outline Generation

Now we can generate the tree-structure outline for the whole slides. Except the title of tag-page, all other texts will be loaded into the outline tree with two hierarchy parameters: the inside-slide-hierarchy (*0∼3, title as 0*) and the slide hierarchy (*1∼3*). One final hierarchy(*1∼6*) will be calculated, but the inside-slide-hierarchy will be also retained, by which we can be more flexible to handle user's demand, for example, when the user wants an outline consisted of titles only.

VI. Evaluation

In our research, we pick up 12 lecture videos from tele-TASK[1] as the dataset to evaluate proposed solution. Tele-TASK videos are stored in double-streams format, one stream focusing on the lecturer and the other recording his/her computer screen. These 2 streams are synchronized and the slides we need can be extracted from the second stream automatically as an initial step of Video OCR Process.And the selected 12 videos belong to 12 different lecturers, by which we reach the maximum possible diversity of the dataset. A lecture ID will be used to indicate each lecture video which can be found online[2].

TABLE I
Statistics about Text-lines

ID	T-Original	T-Modified		T-Final		
		Num	M to O	Num	F to M	F to O
5626	511	243	47.6%	97	39.9%	19.0%
6011	326	148	45.4%	122	82.4%	37.4%
6021	407	219	53.8%	125	57.1%	30.7%
6027	456	344	75.4%	235	68.3%	51.5%
6031	443	253	57.1%	115	45.5%	26.0%
6098	1369	751	54.9%	322	42.9%	23.5%
6102	543	284	52.3%	158	55.6%	29.1%
6104	731	372	50.9%	198	53.2%	27.1%
6106	643	333	51.8%	168	50.5%	26.1%
6196	1029	378	36.7%	175	46.3%	17.0%
6201	434	280	64.5%	140	50.0%	32.3%
6212	1963	1032	52.6%	403	39.1%	20.5%
All	8885	4637	52.4%	2258	48.7%	25.5%

When evaluating, we analyze 2 kinds of entity in our solution, text-lines and slides. By the statistics of text-lines we can mainly test the effect of pre-processing and intra-slide reconstruction which have been illustrated in Section III and IV, while the statistics of slides will tell us how the inter-slides analysis process explained in Section V works. We are not intending to examine the 'accuracy' of the full outline, because proposed solution starts from existing OCR scheme, which has unavoidable accuracy problem itself as we have mentioned many times before. But the accuracy of slide titles will be checked due to its highest importance to the E-Learning portal users, regardless of OCR error.

Table I shows the statistic of text-lines, the basic data structure containing text, in 3 different phases. The T-Original column shows the total number of text-lines achieved from OCR result, and T-Modified contains the stats after text-lines connection (*until Chapter IV-B*), including total number remained and its ratio to the T-Original. T-Final shows all reconstructed text-lines which have been involved in the final tree-structure outline. 'F to M' means the ratio of 'T-Final to T-Modified', while 'F to O' is 'T-Final to T-Original'.

[1]http://www.tele-task.de/

[2]The lecture video with its ID as 'id' can be found in http://www.tele-task.de/archive/lecture/overview/id/

TABLE II
STATISTICS ABOUT SLIDES

ID	S-Initial	S-Final		1st Level		2nd Level		3rd Level		Title Accuracy			
		Num	F to O	Num	Ratio	Num	Ratio	Num	Ratio	N_c	N_p	N_w	R_A
5626	28	18	64.3%	13	64.3%	5	27.8%	0	0.0%	11	0	7	61.1%
6011	18	8	44.4%	8	100%	0	0.0%	0	0.0%	6	1	1	81.3%
6021	29	17	58.6%	9	52.9%	8	47.1%	0	0.0%	15	2	0	94.1%
6027	24	24	100%	12	50.0%	12	50.0%	0	0.0%	20	3	1	89.6%
6031	22	22	100%	15	68.2%	7	31.8%	0	0.0%	10	11	1	70.5%
6098	57	53	93.0%	10	18.9%	43	81.1%	0	0.0%	37	11	5	80.2%
6102	36	29	80.6%	6	20.7%	23	79.3%	0	0.0%	28	1	0	98.3%
6104	37	22	59.5%	17	77.3%	5	22.7%	0	0.0%	17	5	0	88.6%
6106	28	28	100%	6	21.4%	22	78.6%	0	0.0%	23	5	0	91.1%
6196	81	33	40.7%	17	51.5%	16	48.5%	0	0.0%	31	2	0	97.0%
6201	25	24	96.0%	6	25.0%	8	33.3%	10	41.7%	22	2	0	95.8%
6212	89	33	37.1%	8	24.2%	22	66.7%	3	9.1%	26	0	7	78.8%
All	474	311	65.6%	127	40.8%	171	55.0%	13	4.2%	246	43	22	86.0%

From Table I we can easily figure out that nearly 50% original text-lines have been removed or merged before the generation of intra-slide content tree, and before being involved in final outline, almost a further half get eliminated. As a result, only approximate 1/4 of well-organized text will be provided to the users, which ensures that the outline generated is not a simple collection of texts, but still contain enough content to give the users a complete overview of the whole lecture.

Table II shows the statistics about slides in 2 phases. The total number of slides after intra reconstruction is in 'S-Initial' column. And then 'S-Final' shows that how many slides survive after deleting the repeated ones, the 'Live show' or being merged into others. Finally the composition of final slides is revealed by the hierarchical difference.

Over 1/3 of the initial slides have been discarded due to one of three logical reasons we illustrated in Chapter V-B, V-C and V-D respectively, which proves both the importance and the effectiveness of these procedures. And about final slides, over 50% belong to level 2 and even a few in level 3, makes the final outline organized better in a logically progressive manner rather than simply lineal structure.

Table II also shows the accuracy of slide titles in our research. N_c means the number of completely accurate title, while N_p and N_w means the number of partly accurate title and wrong title respectively. And our definition of 'partly accurate' is missing a few words but users can still figure out what the title probably means. Finally a weighted accuracy rate (R_A) has been calculated by following equation:

$$R_A = \frac{N_c + N_p \times 0.5}{N_c + N_p + N_w} \tag{1}$$

Result shows that the average accuracy rate of titles is 86.0% and not a single test video has a lower accuracy rate than 60%, which means at least in title level, the lecture outline we generated is fairly trustworthy.

VII. CONCLUSION

In this paper we propose a solution to generate tree-structure outline for lecture video with synchronized slides by reconstructing text system inside each slide and analyzing logic correlation between slides. The evaluation result shows that the outline retains about only 1/4 of original textual information from OCR process and is organized hierarchically. And the accuracy of slide titles is 86%, which makes the outline fairly reliable to the E-Learning portal users. In the future, we intend to focus on the specialty of different slides type, rather than the common principle, in other words, to make the whole process adaptive.

REFERENCES

[1] B. Erol and Y. Li, "An overview of technologies for e-meeting and e-lecture," in *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*. IEEE, 2005, pp. 6–12.

[2] S. Bateman, C. Brooks, and G. McCalla, "Collaborative tagging approaches for ontological metadata in adaptive e-learning systems," in *Proceedings of the Fourth International Workshop on Applications of Semantic Web Technologies for E-Learning (SW-EL 2006)*. Citeseer, 2006, pp. 3–12.

[3] T. Atapattu, K. Falkner, and N. Falkner, "Automated extraction of semantic concepts from semi-structured data: Supporting computer-based education through the analysis of lecture notes," in *Database and Expert Systems Applications*. Springer, 2012, pp. 161–175.

[4] A. S. Imran, L. Rahadianti, F. A. Cheikh, and S. Y. Yayilgan, "Semantic tags for lecture videos," in *Semantic Computing (ICSC), 2012 IEEE Sixth International Conference on*. IEEE, 2012, pp. 117–120.

[5] M. Grabe and K. Christopherson, "Optional student use of online lecture resources: resource preferences, performance and lecture attendance," *Journal of Computer Assisted Learning*, vol. 24, no. 1, pp. 1–10, 2008.

[6] S. Lonn and S. D. Teasley, "Saving time or innovating practice: Investigating perceptions and uses of learning management systems," *Computers & Education*, vol. 53, no. 3, pp. 686–694, 2009.

[7] H. Yang, F. Gruenewald, and C. Meinel, "Automated extraction of lecture outlines from lecture videos-a hybrid solution for lecture video indexing," in *Proceedings of 4th International Conference on Computer Supported Education (CSEDU)*, 2012, pp. 13–22.

[8] H. Yang, M. Siebert, P. Luhne, H. Sack, and C. Meinel, "Lecture video indexing and analysis using video ocr technology," in *Signal-Image Technology and Internet-Based Systems (SITIS), 2011 Seventh International Conference on*. IEEE, 2011, pp. 54–61.

[9] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," in *Soviet physics doklady*, vol. 10, 1966, p. 707.