Automatic Generation of Learning Channels by Using Semantic Relations among Lecture Slides and Recorded Videos for Self-Learning Systems

Yuanyuan Wang Daisuke Kitayama Ryong Lee Kazutoshi Sumiya
University of Hyogo, Japan
{*nd09s005, ne07p001*}*@stshse.u-hyogo.ac.jp*, {*leeryong, sumiya*}*@shse.u-hyogo.ac.jp*

***Abstract*—We present a method of automatically generating learning channels by using the semantic relations that implicitly exist in slides of a lecture that has accompanying recorded video. These days, many lecture videos with presentation files are shared over the Web from many universities through their own public sites. Although these materials are useful and valuable to many potential students, their use of sequential static media for self-learning purposes means there is still a lack of support for self-learners seeking learning channels suitable for various levels of understanding. Our newly generated learning channels let users easily focus on either highly detailed slides or introductory slides without needing to examine all of the data. We describe a prototype system supported by this learning-channel construction method.**

***Keywords*-multimedia, learning channels, presentation content, semantic relation**

I. Introduction

These days, a lot of lecture materials made from many actual classes in universities or other education organizations are shared on websites such as SlideShare and YouTube EDU [1]. Free online educational contents often consist of presentation slides and recorded video. Thus, not only students who missed the class but also any other people interested in the topic can review the class and study the content by themselves later. However, compared with actual participation, learning throughout such achieved material is more passive and tedious because of the lack of interactivity and intensity; the dominant learning style for online presentation contents is just viewing sequentially arranged slides and video. Such unidirectional injective learning cannot easily attract a self-learner's interest and requires the user's effective attention. On the other hand, making the static contents much more dynamic and interactive would require a lot of effort by lecturer. To fill the gap between the lecturers' limitations and students' diverse requirements in practice, we propose a dynamic reorganization of the almost raw contents, which are easily available on the Web but unable to meet the needs of students having various levels of understanding.

Reorganizing presentation contents to suit users' interests or capabilities could be achieved mainly by (1) summarizing long contents into short intensive highlights [1] that include what users need to know and (2) constructing a hierarchical or graph-like structure like HTML documents [2] without keeping the sequential ordering, but focusing on relations among slides or video segments. However, the first approach often fails to cope with dynamic changes in users' interests because once the highlights have been created, they are hard to reorganize according to given summarization criteria. Unlike the relatively static approach, the second approach is very flexible at supporting dynamic changes in users' interests during the learning processes, but there are generally no special linkages explicitly represented in either slides or videos. Thus, it would be necessary to translate presentation contents into dynamic and semantic learning channels, where each student is supported throughout dynamic changes in his or her interests according to his/her learning level by using the semantic relations among slides that has accompanying video segments from archives. The concept of our approach is shown in Figure 1.

The remainder of this paper is organized as follows. Section 2 describes our approach and related work. Section 3 addresses the mathematics for determining semantic relation types. Section 4 explains the extracted scenes presented by our learning-channel construction method and discusses our prototype system.

II. Approach and Related Work

A. Motivating Example

First, we give an example that illustrates why we were motivated to conduct this research. Suppose that a lecture (combination of slides and corresponding video) introduces apples and oranges (see the upper part of Figure 2). The slide titles are as follows in scene order: {*Fruits*, *Apple*, *Orange*, *Sugar*, *Sour apple*, *The apple*}. If the user is interested in the *Apple* slide, he or she can use a learning-channel engine to search for relevant scenes about it, such as those in the lower right of Figure 2. If *Apple* explains the "components of apple", it is more likely to be a general-content scene than a detailed explanation such as the "kinds of sugar in an apple" in *Sugar*. Indeed, *Apple* contains "sugar and sour components", so it is similar in content to *Sour apple*. On the other hand, the explanation provided by *Apple* is more likely to have detailed content about apples than the explanation about "apple" in *Fruits*. *Orange* covers a different topic, i.e., "oranges", so we think that the speaker's additional

[1] http://www.slideshare.net/, http://www.youtube.com/edu/

978-0-7695-3890-7/09 $26.00 © 2009 IEEE
DOI 10.1109/ISM.2009.100

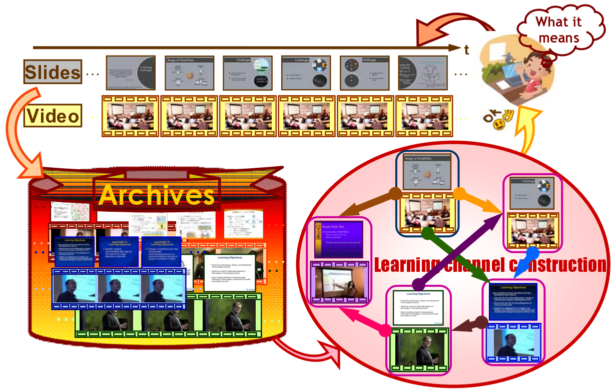

Figure 1. Concept of our approach

comments about "apple" describe the "sugar and sour components" in *Orange*. If one searches for scenes by the conventional method using matching keywords, keywords tend to appear many times in a series of scenes. A lot of extracted scenes contain the same keywords, but irrelevant scenes are often included, like those in the lower left of Figure 2. Thus, it is useful to use the relations among scenes about *Apple* and the user can easily understand the context of *Apple* well without looking at the whole lecture.

However, we think that extracted scenes cannot narrow down the topics by using only relations. When the user is interested in *Apple*, it is effective to use a relation to search for scenes that provide more information than *Apple* does. *Sugar* is more likely to have detailed content about the "kinds of sugar in an apple" than "apple's components and kinds" in *Apple* is. *The apple* contains information about "apple" and "sour" in *Apple*, and *Sour apple* is more likely to have detailed content about "components of sour apple" than "kinds of apple including sour apple" in *The apple* is. Meanwhile, *Sour apple* is more likely to have detailed content about "components of sour apple" than "apple's components and kinds" in *Apple* is. Thus, these scenes are related to *Apple* and can better help the user's in understanding his/her topic of interest.

Consider the case where the user selects *Apple* and *Sugar* because of interest in "sugar and sour". In this case, *Sugar* is more likely to have details about the content in *Apple*, so the relation from *Apple* to *Sugar* can be estimated. Furthermore, *The apple* contains information about the content of *Apple*. And the scene pair, [*The apple*, *Sour apple*] is extracted because *Sour apple* is more likely to have details about the content in *The apple* because the relation from *The apple* to *Sour apple* is the same as the [*Apple*, *Sugar*] relation. Thus, the user can select a pair of scenes. Our method estimates the relation between the two scenes in this pair and then searches for all scene pairs that have the same relation and presents them to the user. To help users understand better, it must extract the scene pairs by using their semantic relation.

B. Our Approach

We propose a method of automatically generating learning channels with a differential base by analyzing specific relations. We extract structural information such as indents and the logical set of text in slides and the keyword set for text in the speech in a video and provide this information so that our method can use it (see Figure 3). The content is divided on the basis of the speaker's slide changes. Each slice produced by the division is called a scene, which consists of *one* slide and *one* video segment containing a recording of the speaker's explanation of that slide.

We defined the structural information on the basis of indents in slide text. The slide title (1st level indent) is the upper level. The first item of text is on the 2nd level and subitems deepen with the level of indentation (3rd level, 4th level, and so on). Indents outside text such as figures or tables are on the average level for the slide. It is usual that the lower-level indented keywords are supplementary and explain the upper-level keywords. We define semantic relations between scenes by using the metadata of slides and videos, for example, a certain scene is a much more detailed explanation than other scenes.

Our basic method is to extract corresponding scene pairs by using the relation between the two scenes of the selected pair. Our approach involves two types of input: (1) selecting one scene and one relation for which the user wants to get corresponding scene pairs for some purpose and (2) selecting a scene pair and the relation between the two scenes for which the user wants to get other scene pairs related to the selected scene pair. First, we determine semantic relations by examining how a keyword's indent position varies in different slides and how frequently the keyword appears the video. As a result, users can get semantic scenes as learning channels from inputs that may or may not be contiguous.

C. Related Work

Smith et al. [3] proposed a method of extracting video segments from a video by detecting the features of scenes by analyzing their CC (Closed Caption), color, and speech features. Yokota et al. [4] proposed a system called Unified Presentation Slide Retrieval by Impression Search Engine (UPRISE) for retrieving a sequence of desired presentation slides from archives of combinations of slides and video. Nakano et al. [5] proposed a method of using laser pointer information in lecture scene retrieval by UPRISE. Le et al. [6] proposed a method of extracting important scenes for automatically generating digests from presentation videos recorded using MPMeister II [2]—a tool for multimedia web contents. Their method extracts important scenes from unified content based on the metadata features of a single medium or two heterogeneous media. Our goal is to extract the relations between scenes on the basis of the metadata

[2]http://www.ricoh.co.jp/mpmeister/

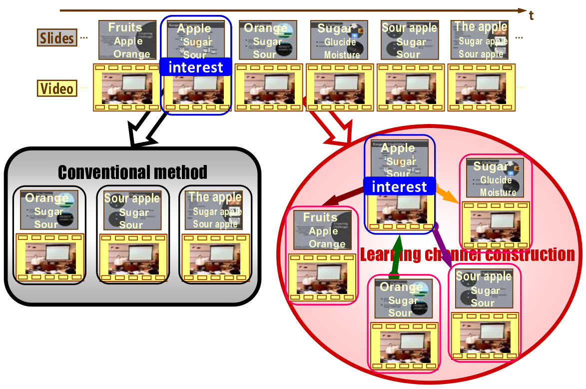

Figure 2. Results obtained by conventional method and by learning channels

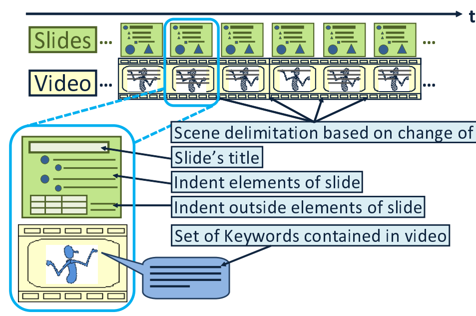

Figure 3. Metadata of unified content

features of heterogeneous media for various applications to support users' self-studies effectively.

Pradhan et al. [7] proposed a gluing operation that generated a new video segment from a set of video segments. This method was targeted at a single medium such as video, so their target was different from the target of our method for extracting scenes from unified content consisting of heterogeneous media, such as videos with presentation files.

Kan [8] proposed a method of storing aligned presentation and document pairs. This method synchronized presentation slides and document paragraphs on the basis of text similarity. On the other hand, our aim is to integrate heterogeneous media such as slides and videos using relations existing in slides with recorded videos.

Kitayama et al. [9] proposed a method of extracting scenes according to their relations and roles. This research is similar to ours as a method for using the relations between scenes. However, we think that using semantic relations to extract corresponding scene pairs from unified contents helps users to understand the context more easily.

III. Using Semantic Relations of Learning Channels

A. Judgment of Semantic Relation Types

We define the selected scene of interest as the basic scene and define other scenes that have specific relations as being semantically related to the basic scene through one of four semantic relation types: *detailed*, *generalized*, *similar*, and *additional* (see Figure 4). Scenes that have semantic relations are called semantic scenes, i.e., if a scene has a detailed relation, we call this scene a detailed scene.

Detailed and generalized scenes are functionally interchangeable, while a basic scene is a generalized scene from the viewpoint of a detailed scene.

This section explains how the semantic relation type is determined. Let a_i be the slide number of a basic scene and a_j be the slide number of a scene we want to detect. The semantic relation types are determined for all scenes.

B. Judgment of Relation Type as Detailed or Generalized Relation

If a scene has more information than the basic scene, its relation to the basic scene is *detailed*. The slide of a scene that contains specialized content is talked about at greater length by the speaker. If a scene contains content in the outline given in a *generalized* scene, it is described in relation to the basic scene. The slide of a scene that contains generalized content is talked about less by the speaker. Because *detailed* and *generalized* scenes are equivalent, we explain only the determination of *detailed* scenes by using keywords present in basic scene a_i and the scene to detect a_j here.

$$|U(a_i,a_j)| > |S(a_i,a_j)| \quad (1)$$

$$|U(a_i,a_j)| > |D(a_i,a_j)| \quad (2)$$

$$\frac{v\_n(U(a_i,a_j),a_i)}{v\_c(a_i)} < \frac{v\_n(U(a_i,a_j),a_j)}{v\_c(a_j)} \quad (3)$$

If the levels conform to Eqs. (1) and (2), and the ratio of keywords in the video conforms to Eq. (3), then a_j is determined to be detailed. This is because the keywords in the a_jth slide appear more frequency than in the a_ith slide, and the a_jth video segment contains many keywords. $U(a_i,a_j)$ is the set of keywords in levels that ascend from the a_ith slide to a_jth slide. $S(a_i,a_j)$ is the set of keywords in the same level in both the a_ith and a_jth slides. $D(a_i,a_j)$ is the set of keywords in levels that descend from the a_ith slide to a_jth slide. If the number of keywords in $U(a_i,a_j)$ is extracted more frequently than the number in $S(a_i,a_j)$ and $D(a_i,a_j)$ in Eqs. (1) and (2), then the level of the slide is judged to be higher. In Eq. (3), $v\_n(U(a_i,a_j),a_i)$ is the number of keywords in $U(a_i,a_j)$ in the a_ith video segment and $v\_c(a_i)$ is the total number of keywords in the a_ith video segment.

C. Judgment of Relation Type as Similar Relation

If the slide of a scene contains similar content to the basic scene and its video segment has a similar quantity of speech

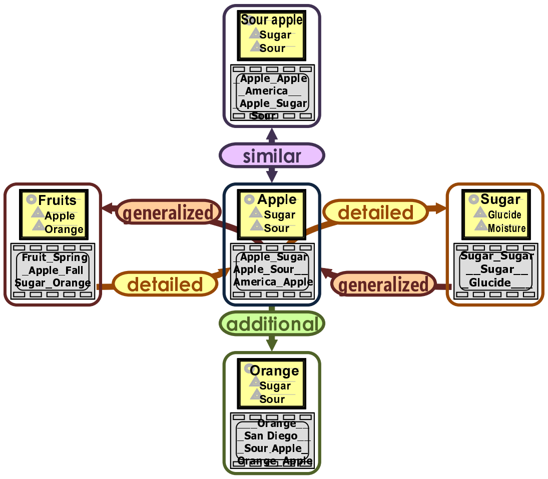

Figure 4. Semantic relation types

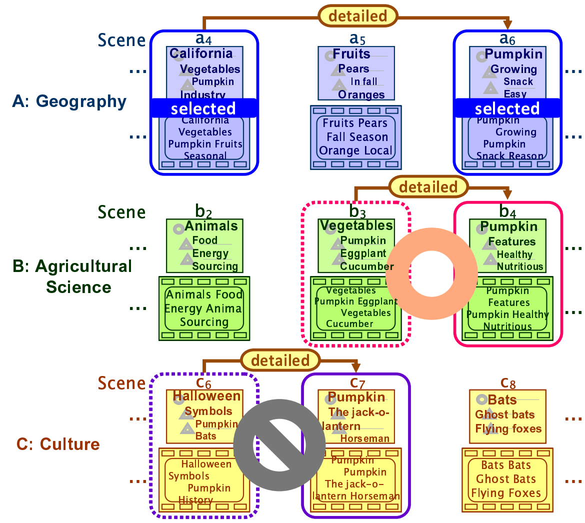

Figure 5. Scene pairs extracted by detecting a corresponding scene

to that of the basic scene, then the relation between these scenes is *similar*.

$$|S(a_i, a_j)| > |U(a_i, a_j)| \quad (4)$$

$$|S(a_i, a_j)| > |D(a_i, a_j)| \quad (5)$$

$$\left| \frac{v\_n(S(a_i, a_j), a_i)}{v\_c(a_i)} - \frac{v\_n(S(a_i, a_j), a_j)}{v\_c(a_j)} \right| < \alpha \quad (6)$$

If the levels conform to Eqs. (4) and (5), and the keyword ratio in the video conforms to Eq. (6), then a_j is determined to be similar. This is because the keywords in the a_jth slide appear with a similar frequency in the a_ith slide, and the keyword ratio is similar in both the a_ith and a_jth video segments. If the number of keywords in $S(a_i, a_j)$ is extracted more frequently than the number in $U(a_i, a_j)$ and $D(a_i, a_j)$ in Eqs. (4) and (5), then the hierarchical structure of the slide is determined to be the same. In Eq. (6), $v\_n(S(a_i, a_j), a_i)$ is the number of keywords in $S(a_i, a_j)$ in the a_ith video segment and α is a threshold.

D. Judgment of Relation Type as Additional Relation

If a scene contains another topic related to the basic scene, its relation to the basic scene is *additional*. The speaker's additional comments can also be used to describe the content in other scenes. The speaker descriptions also include the keywords contained in the slide of the basic scene. This *additional* scene helps users to understand the basic scene by providing extra information.

$$|inter(a_j, a_i)| < |differ(a_j, a_i)| \quad (7)$$

$$\frac{s\_n(inter(a_j, a_i), a_i)}{s\_c(a_i)} > \frac{s\_n(inter(a_j, a_i), a_j)}{s\_c(a_j)} \quad (8)$$

$$v\_n(inter(a_j, a_i), a_i) > 0 \quad (9)$$

$$v\_n(inter(a_j, a_i), a_j) > 0 \quad (10)$$

$$l(k_x, a_i) < \frac{l\_lowest(a_i)}{2} \text{ or } l(k_x, a_j) > \frac{l\_lowest(a_j)}{2} \quad (11)$$

If the levels conform to Eqs. (7) and (11), the keyword ratio in the slide conforms to Eq. (8), and the keywords in the video conform to Eqs. (9) and (10), then a_j is determined to be additional. This is because there is an explanation of the a_ith scene in the a_jth scene, and there is more explanation in the a_ith slide. Let $inter(a_j, a_i)$ be the set of keywords that appear in both the a_ith and a_jth scenes and let $differ(a_j, a_i)$ be the set of keywords that do not appear in a_ith scene. If the number of keywords in $differ(a_j, a_i)$ is more than the number in $inter(a_j, a_i)$ in Eq. (7), then the keywords in $inter(a_j, a_i)$ are common keywords in both the a_ith and a_jth scenes. In Eq. (8), $s\_n(inter(a_j, a_i), a_i)$ is the number of common keywords in the a_ith slide and $s\_c(a_i)$ is the total number of keywords in the a_ith slide. In Eq. (9), $v\_n(inter(a_j, a_i), a_i)$ is the number of common keywords in the a_ith video segment. In Eq. (11), $k_x \in inter(a_j, a_i)$, function l_lowest is the lowest level of the slide, so we can estimate whether common keywords ascend from a lower level to an upper level.

E. Types of Input

IV. Automatic Generation of Learning Channels

Our learning-channel construction method involves two types of input: (1) by selecting one scene and one relation, the user wants to get corresponding scene pairs for some purpose and (2) by selecting a scene pair, the user wants to get corresponding scene pairs that are related to the selected scene pair. When one scene is selected, our system extracts the corresponding scene pairs that have the selected type of semantic relation for the user's interest by detecting scenes that correspond to the selected scene. When the user selects a scene pair of interest, our system determines its relations with all other scene pairs and extract ones judged to be corresponding ones.

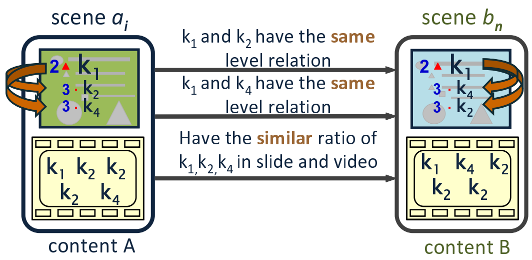

Figure 6. Determination of corresponding scene

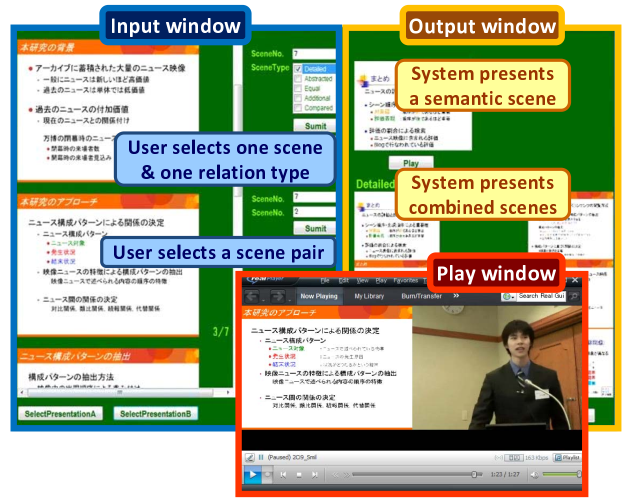

Figure 7. Screenshot of prototype system

A. Detection of the Corresponding Scenes

The contents of three lectures are shown as examples in Figure 5. When a user studying geography (A) is interested in scenes a_4 and a_6 about "pumpkin as a vegetable", the scene pair [a_4, a_6] is selected, and the relation from a_4 to a_6 to be *detailed*. In this case, we consider that it would be useful for the user if the system presented him/her with other detailed scene pairs having relevant points from other lectures. [b_3, b_4] in B explains about "pumpkin as a vegetable", and b_4 explains in more detail than b_3. Thus, [b_3, b_4] also corresponds to the selected scene pair. On the other hand, although c_7 provides more details about "pumpkin" than c_6 in C, these scenes are not related to the selected scenes because they explain about the pumpkin as a symbol of Halloween. Therefore, [b_3, b_4] is extracted as a corresponding scene, where b_3 is a scene in other contents that corresponds to the basic scene a_4 of the selected scene pair. Thus, in extracting corresponding pairs, it is necessary to refer to the corresponding scene.

Let a_i be a basic scene in content A, and let b_n be a candidate scene in content B. If the keywords in a_i and b_n satisfy the following conditions, then the scene b_n is judged to be a corresponding scene in other contents (see Figure 6).

$$K(a_i) = \{\{k_x, k_y\} | l(k_x, a_i) < l(k_y, a_i)\} \tag{12}$$

$$\frac{|K(a_i) \cap K(b_n)|}{min(|K(a_i)|, |K(b_n)|)} > \beta \tag{13}$$

$$SVratio(a_i, b_n) = \frac{s\_n(K(a_i) \cap K(b_n), a_i)}{v\_n(K(a_i) \cap K(b_n), a_i)} \tag{14}$$

$$|SVratio(a_i, b_n) - SVratio(b_n, a_i)| < \gamma \tag{15}$$

In Eq. (12), $K(a_i)$ is the set of keyword pairs, where the level of keyword k_x is higher than the level of keyword k_y in the a_ith slide. In Eq. (13), $| K(a_i) \cap K(b_n) | / min(| K(a_i) |, | K(b_n) |)$ calculates the degree of the hierarchical relation between the keywords in the a_ith and b_nth slides, and β is a threshold. In Eq. (14), function $SVratio(a_i, b_n)$ calculates the degree of the number of $K(a_i) \cap K(b_n)$ keeping the hierarchical relation in the a_ith slide and video segment. $s\_n(K(a_i) \cap K(b_n), a_i)$ is the number of $K(a_i) \cap K(b_n)$ in the a_ith slide, and $v\_n(K(a_i) \cap K(b_n), a_i)$ is the number of $K(a_i) \cap K(b_n)$ in the a_ith video segment. In Eq. (15), the function calculates the similarity of $K(a_i) \cap K(b_n)$ in a_i and b_n, and γ is a threshold.

B. Generation of Learning Channels

Learning channels extract corresponding scene pairs determined by selecting a scene pair that has the same semantic relation between its scenes. They produce different outputs depending on the type of input (there are two types of input).

1) Determination of Learning Channels by Inputting One Scene: When a scene is selected, the system searches for scenes in other contents using the selected relation type. We think that the user understood the selected content but wants to gain a better understanding of the topic.

In Figure 5, after the user understood A, he selected a_4 and *detailed* in order to study a_4 and gain a more *detailed* understanding of it. a_6 explains a_4 in detail, but it is also useful to present the user with relevant detailed scenes in other lectures. b_3 corresponds to a_4. The corresponding scene pair [b_3, b_4] in B explains about "pumpkin as a vegetable", and b_4 explains in more detail than b_3. So, b_4 is a detailed explanation of the content in a_4 that could help the user to understand "pumpkin as a vegetable" in detail by utilizing content from other lectures. Therefore, we think that the user understood a_6 in A so it is not presented, extracting only b_4 can satisfy the user's demand.

2) Determination of Learning Channels by Inputting a Scene Pair: When a scene pair is selected, our method estimates all relations between it and corresponding scene pairs in other lectures. We think that if the user looks at only the content of a single lecture, he/she can understand it well with a little supplementary explanation from other lectures.

In Figure 5, when the user selects [a_4, a_6] out of interest in "pumpkin as a vegetable", a_6 explains a_4 by providing more detail, so our method estimates the relation from a_4 to a_6 to be *detailed*. In this case as well, we think that it is useful for the user to be presented with relevant detailed scenes from other lectures. b_3 corresponds to a_4. The corresponded scene pair [b_3, b_4] in B explains about "pumpkin as a vegetable", and b_4 explains in more detail than b_3. So, b_4 is treated as a supplementary explanation of the selected scenes and can help the user to understand "pumpkin as a vegetable". Therefore, extracting b_4 combined with [a_4, a_6] can help the user to understand his/her topic of interest.

C. Prototype System

We have developed a prototype system to support the learning-channel construction engine (see Figure 7) in Microsoft Visual Studio 2008 C#. This prototype implements the determination part and the output part. In the determination part, all semantic relation types are determined using the video and slide metadata, and scenes corresponding to the basic scene are detected. The corresponded scene pairs are determined by using the semantic relations and the corresponding scene. The terms in the slides and video are extracted using the morphological analyzer *Mecab* [10], which is in SlothLib [11].

A list of slides and scene numbers is displayed in the input window. The user can select scenes of interest by inputting the scene numbers in the textbox and by checking the semantic relation type in the list. If the user selects either (1) a scene and a relation or (2) a scene pair, then slides of semantically related scenes are presented in the output window. The code for controlling the output in other windows is described in the Synchronized Multimedia Integration Language (SMIL).

V. Concluding Remarks

We have proposed a learning-channel construction engine that uses semantic relations. It automatically generates learning channels to extract scenes and combined scenes from unified contents based on semantic relations. The type of semantic relation is determined on the basis of the metadata of structural information, such as indents and texts in slides, and the set of keywords in the text of the speech in the video. Thus, users use the learning-channel engine to look for scenes that have relevant points to the scene of interest or appropriate combined scenes from unified contents. This approach is very effective.

We have also developed a prototype system and evaluated it using actual presentation data. We confirmed an improvement in the coverage of semantic relation types and their definition and in the detection of corresponding scenes for extracting corresponding scene pairs effectively by using the semantic relations.

In the future, we want to improve the algorithm for determining the semantic relations and corresponding scenes and evaluate our approach with a large set of actual educational contents. Furthermore, our method could extend the range of available educational materials that would be useful if other related content, such as related papers, graphics, and Internet content, were also unified.

Acknowledgment

This research was supported in part by a Grant-in-Aid for Scientific Research (B)(2) 20300039 from the Ministry of Education, Culture, Sports, Science, and Technology of Japan.

References

[1] C. Choudary and T. Liu, "Summarization of visual content in instructional videos," in *IEEE Transactions on Multimedia*, vol. 9, 2007, pp. 1443–1455.

[2] A. Mehler, E. Mehler, M. Dehmer, and R. Gleim, "Towards logical hypertext structure - a graph-theoretic perspective," in *Proc. of I2CS '04, Guadalajara/Mexico, Lecture Notes in Computer Science, Berlin-New*. Springer, 2004.

[3] M. Smith and T. Kanade, "Video skimming and characterization through the combination of image and language understanding techniques," in *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR07)*, 1997.

[4] H. Yokota, T. Kobayashi, T. Muraki, and S. Naoi, "Uprise : Unified presentation slide retrieval by impression search engine," in *IEICE Transactions on Information and Systems*, vol. E87-D, no. 2, Feb. 2004, pp. 397–406.

[5] W. Nakano, T. Kobayashi, Y. Katsuyama, S. Naoi, and H. Yokota, "Treatment of laser pointer and speech information in lecture scene retrieval," in *Proc. of the Eighth IEEE International Symposium on Multimedia (ISM2006)*, Dec. 2006, pp. 927–932.

[6] H.-H. Le, T. Lertrusdachakul, T. Watanabe, and H. Yokoda, "Automatic digest generation by extracting important scenes from the content of presentations," in *Proc. of the 19th International Conference on Database and Expert Systems Application (DEXA2008)*, Sep. 2008, pp. 590–594.

[7] S. Pradhan, K. Tajima, and K. Tanaka, "A query model for retrieving rele-vant intervals within a video stream," in *Proc. of the 6th IEEE Int'l Conference on Multimedia Computing and Systems (ICMCS 99)*, vol. 2, 1999, pp. 788–792.

[8] M.-Y. Kan, "Slideseer: a digital library of aligned document and presentation pairs," in *Proc. of the 7th ACM/IEEE-CS joint conference on Digital libraries*, 2007, pp. 81–90.

[9] D. Kitayama, A. Otani, and K. Sumiya, "A scene extracting method based on structural and semantic analysis of presentation content archives," in *Proc. of the 7th International Conference on Creating, Connecting and Collaborating through Computing (C5 2009)*, Jan. 2009.

[10] MeCab, http://mecab.sourceforge.net/.

[11] SlothLib, http://www.dl.kuis.kyoto-u.ac.jp/slothlib/.

[12] M. Asahara and Y. Matsumoto, "Extended models and tools for high-performance part-of-speech tagger," in *Proc. of the 18th International Conference on Computational Linguistics (COLING 2000)*, 2000, pp. 21–27.