VIDEO INDEXING AND AUTOMATIC CAPTION CREATION

ABSTRACT

This paper presents the design and implementation of a video indexing and automatic caption creation system. The system is able to extract audio from videos and to get the transcript directly from the audio file using the newly designed audio-to-text engine based on Hidden Markov Model (HMM). Transcripts can be edited and the corresponding time stamps are updated automatically. The video indexing feature is capable of searching videos using keywords from the speech. If a match is found, the video player jumps to the keyword position and starts to play the video clip. The system is capable of supporting more than 40 common video formats. In addition, all videos can be streamed using the YouTube style streaming service we developed. The resolution of video is adjusted automatically according to the network speed of the client.

KEYWORDS

Video indexing, automatic caption, time stamp calculation, video streaming

1. INTRODUCTION

Text-based search is very common on websites nowadays. Search engines can search static text, they can also search text in Word or PDF documents. Some custom designed search engines are able to search dynamic content such as data stored in the database. The ttaconline search engine is the one we developed that can search web pages, Word and PDF files and dynamic content. The algorithm has been published by the authoritative organization of modern web – World Wide Web Consortium (W3C).

With the rapid growth of media content especially video content in education and in many social networks, there is a need to understand videos even before a video is played. Automatic transcript extraction is an important step toward understanding videos. Once a video transcript is created, it can then be used as caption and further be indexed. Indexed videos are searchable and can be played directly from the keyword if the search yields a result.

2. AUTOMATIC TRANSCRIPT EXTRACTION

The system is a standard C# application together with a web application. It consists of database servers, web application servers, personal computers, wireless networks and handheld devices such as PDA's and cell phones.

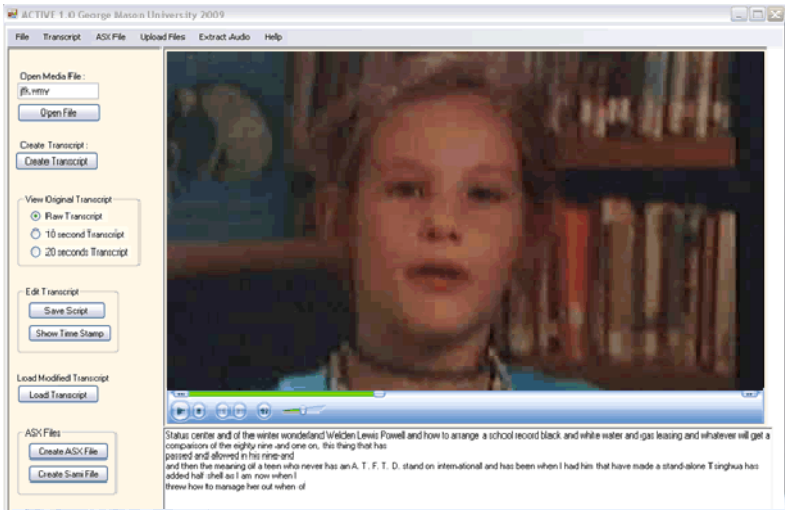

Fig. 1. Video Caption Creation

The automatic transcript extraction works in two steps. When a video is loaded into the system, the audio extraction agent is automatic activated and an audio file is produced. This process takes only a tiny portion of the video time by using the fast audio extraction algorithm we developed. Once the audio file is generated, the audio-to-text agent starts looking through the audio file, finds the timestamp of each word and extracts text. Hidden Markov Model (HMM) is used in the video segmentation process. This process takes longer but it much less than the duration of the video. Figure 1 is the interface of the application.

The system backend (server side) runs with multiple intelligent agents [Wang, 08] (Fig. 2) to 1) ensure system and data security; 2) upload and encode videos; 3) automatic video caption creation; 4) index and search videos; and 5) create video quizzes and evaluate the effectiveness of learning. These intelligent agents run in the background to assist users for their study.

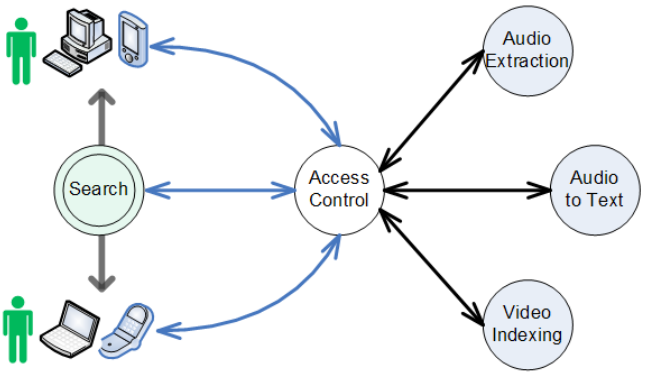

Fig. 2. Intelligent Agents

3. TIME STAMPS CALCULATION

When a speech recognition engine completes recognition of an audio stream, it generates a list of words along with related timing information: time offset and duration. The information is stored in a XML file. The sum of time offset and duration of one word equals to the time offset of the next word.

Speech engines accelerate recognition speed by dividing an audio stream into several audio segments, and the time offset for each segments is the local offset within each segment. So we need to use the original time offset and duration to calculate the time offset of each word from the beginning of the audio stream. Here we call this offset the global time stamp.

Due to the limitation of the speech engine, the accuracy of the recognition cannot be perfect. Thus, transcript editing is necessary. Editing usually include adding, deleting or modifying words. Time stamps calibrations are needed for each of the editing operation.

Figure 3 illustrates the modification and deletion process. We use a link table to store the current row of words. The original XML is shown as Fig. 3a. When the word “ning” is deleted from the link table, the program find a match in the XML file and subtract the durations in the time stamp from the succeeding words. If more than one word is deleted at the same time, then the pointer i will increase until a match is found. (Fig. 3b)

When word “mine” is changed to “mining”, we simply update the word in the XML file without changing the time stamp. (Fig. 3c)

		i →						
1	2	3*	4	5	6	7	8	9
Data	mine	**ning**	is	to	discover	pattern	from	data

(a) Original XML file

		→ **j**					
1	2	3	4	5	6	7	8
Data	mine	is	to	discover	pattern	from	data

(b) Link table after the word "ning" is removed

	j						
1	2	3	4	5	6	7	8
Data	**mining**	is	to	discover	pattern	from	data

(c) Link table after the word "mine" is updated

Figure 3 Time Stamp Calculation

To insert a word in the transcript, we assume that the inserted word will not affect the time stamp of the subsequent words. We add "null" as both the time offset value and the duration.

4. VIDEO INDEXING

Video indexing is an important step to make video searchable. During the automatic transcript extraction process, every word in the transcript is time-stamped. When a keyword is inputted into the system, the video indexing agent first looks through the transcript finding out the matches. Then it directs the video player jump to the corresponding starting point to play. The algorithm calculates the occurrence of a word as the weight of the keyword (term).

To represent the weight $w_{i,j}$ measuring the relative importance of each concept or single-term T_j, j=1, 2, …, t, in a document D_i, we use the following formula:

$$w_{ij} = tf_{ij} \cdot idf_j$$
$$idf_j = \log\left[\frac{n}{df_j}\right]$$

where tf_{ij} is the frequency of the term T_j in the document D_i, n is the number of documents, df_j is the number of documents in which T_j occurs, and idf_j is the inverse document frequency.

To account for document length, the system normalize both tf and idf components to the range [0, 1]. By dividing tf by the maximum tf value for any term in the document space, we got:

$$w_{ij} = ntf_{ij} \cdot nidf_j \ ,$$
$$ntf_{ij} = \frac{tf_{ij}}{\max\ tf_i},$$
$$idf_j = \log\left[\frac{n}{df_j}\right]$$

After the calculation, all words in the transcript are weighted and is ready to be searched. The relevant of the search results is ranked by the weight value assigned at this point.

5. CONCLUSIONS AND FURTHER DISCUSSIONS

By extracting transcript from video automatically, the system enables fast and effective video index creation. Therefore videos can be searched by any words in the speech. The educational significance is that the system can be used for deaf and hearing-impaired people to study educational material created in video format. It can also be used to create video quizzes.

After videos are indexed, every word in the transcripts is attached with a timestamp. When search keyword is inputted, system looks through the transcript and finds the perfect match. In the video view, the player jumps to the keyword directly and starts playing from that point. At the same time, the keyword is highlighted.

A video quiz can be created by assigning the first word and last word of a phase in the text (transcript). Then a label (question) is attached to the phase. The quiz is later showed on the screen followed by the corresponding cropped video as the answer.

Podcasts are very common and convenient way of learning. People with iPod or other handhelds can use their devices to download m-Learning content and to study it whenever convenient or wherever possible. With the templates we designed, users can also study the video material with their handheld device anywhere at any time.

ACKNOWLEDGEMENT

This ongoing research is funded by U.S. Department of Education under a grant [H327A080013].

REFERENCES

Snoek, C.G.M., and Worring, M. 2002. A Review on Multimodal Video Indexing. In *Proceeding of IEEE International Conference on Multimedia and Expo.* Vol. 2, pp. 21-24

Panchanathan, S., Mandal, M.K., and Aboulnasr, T. 1998. Video Indexing in the Wavelet Compressed Domain. In *Proceeding of International conference on Image Processing*. Vol. 3. pp. 546-550

Kozina, B., et at, 2006. Agent-based Messaging System for M-Learning. In *Proceeding of IEEE MELECON*, May 16-19, pp. 1213-1216.

Lopes, R. F., and Cortes, O. A. C., 2007. A Uniquitous Testing System for m-Learning Environments. In *Proc. of International Conference on Systems and Networks Communications*.

Hentea, M., 2004. *Multi-Agent Security Service Architecture for Mobile Learning*. Purdue University Calumet, pp. 91-95.

Anjos, L. S., and Oliveira, J. M. P., 2006. A Navigation Interface for Adaptive m-Learning Applications. In *Proc. of the Sixth International Conference on Advanced Learning Technologies.*

Liu C., Chen, A.L.P. 2002. 3D-List: A Data Structure for Efficient Video Query Processing. IEEE Transactions on Knowledge and Data Engineering. Vol. 14. Iss. 1. pp. 106-122.

Wang, S., and Behrmann M. 2008. Agent-based Ubiquitous m-Learning Portal for K-12 Teachers, *IEEE/ACM International Workshop on Context-Aware Mobile Learning (CAML)*, pp. 525-529.

Wang, S. Chen, J. and Behrmann M. 2004. Visualizing Search Engine Results of Data-Driven Web Content, In *Proc. of W3C Web Accessibility Initiative*