Aligning Words from Speech Recognition and Shots for Video Information Retrieval

Yu-Jen Cheng and Hsin-Hsi Chen
Department of Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan
E-mail: yjcheng@nlg.csie.ntu.edu.tw; hhchen@csie.ntu.edu.tw

Abstract

In this paper we propose an approach to improve video information retrieval by finding the relationships between ASR tokens and the high-level feature tokens. We employ WordNet hierarchical relationship to calculate word-to-word distances, and align the related words in ASR to the corresponding shots. Finally we define a representative document for each shot, which contains the original ASR, high-level features and the aligned words. Adopting Okapi as our IR system to access the representative documents, we tested different parameters on TRECVID 2003 search task and choose two sets of parameters to test on TRECVID 2004 task (*M_C_2_NLP_Lab1_1* and *M_C_2_NLP_Lab1_2*) along with one ASR baseline run (*M_C_1_NLP_Lab1_1*). The experimental results show that in some cases our algorithm does resolve the time-delay problem, so that the IR performance is improved. In those cases the news scripts talk about an *event*, and do not contain words related to the *object* we want to find, our alignment algorithm does not have significant effects. In summary, the alignment algorithm can improve the overall video IR performance when the high-level features are annotated.

1 Introduction

Video information retrieval (IR) differs from traditional text IR in many ways. Cues from different sources like subtitles, audios, videos, and so on, may be employed. Many video low-level features can be discovered, extracted, and/or combined together to bring in information for retrieval [1]. Some applied automatic image annotation on key frames of each shot to improve the video IR performance [2]. Some use statistics to calculate the co-occurrences of visual tokens and words for each shot inside the story segments [3]. Comparing the low level features, subtitles and audios contribute more semantic information after some preprocessing such as video OCR and automatic speech recognition (ASR). Intuitively, a text-based IR system can be adopted after such kinds of transformation. However, there is a time decay problem between text and shots. We can imagine a scenario in which an anchorperson or a reporter reports some news story, and then the scene changes to the relating locations, objects, or persons. If we use raw ASR text to perform text retrieval, we often retrieve a wrong shot with an anchorperson/reporter introducing news stories. Because of the time-delay problem in the shots, we cannot apply the text information directly.

This paper proposes a method to improve the video IR performance. Given the high-level feature annotation, which represents the shot key frame features, and the original ASR results, we try to set up a correspondence between shots and ASR words using WordNet. A distance measure is proposed to align the ASR words and the shots. This paper is organized as follows. Section 2 introduces the overall system

architecture. Section 3 shows our experimental results in the TRECVID 2004 search task and some discussions. Section 4 concludes the remarks.

2 System Architecture

In news video, a speaker often mentions some place, object, scene or person, and then the shots containing the mentioned item appears on the screen after some decay. Figure 1 shows an example, where S_i denotes the i^{th} shot, and ASR_i and $COMF_i$ denote its ASR outputs and high-level features, respectively. The speaker says, "… buried in the *tomb* of …" at shot S_{i-1}, but the shot containing the images of *tomb* is S_{i+1}. If the ASR text is used to represent a document of a shot, the shot retrieved would be S_{i-1}.

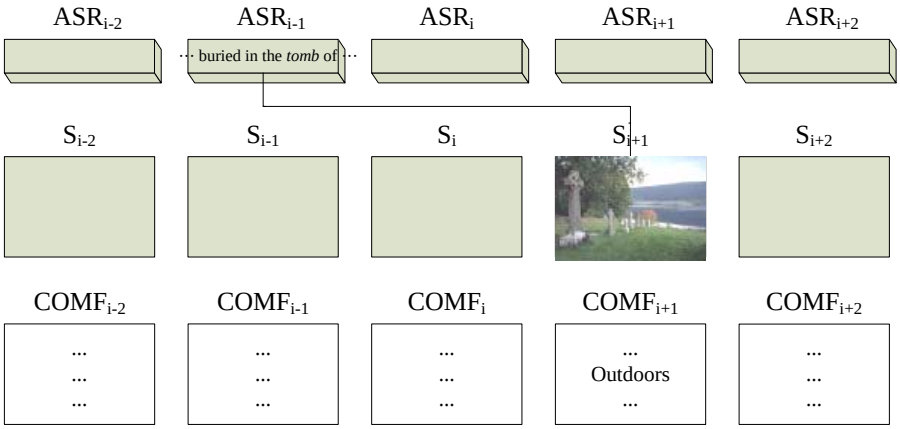


Figure 1. The Time-delay Problem in Video IR

To deal with this problem, we propose a method to find out the relationship between ASR words and high-level feature annotations (COMF) words. In this way, the right ASR words to their corresponding shots, i.e., the shots containing the images of ASR words, will be aligned. In Figure 1, one of the high-level feature annotations "*Outdoors*" gives a hint, which tells us that there is an outdoor scenario and it is likely that the shot contains a *tomb* scenario.

2.1 Extracting Tokens

First we extract the ASR tokens of each shot. The Eric Brill's part-of-speech tagger [4] is used to find out all nouns (called ASR tokens hereafter) of the ASR results. The selection of nouns is due to the assumption that the annotated high-level features, which might be mentioned earlier by the speaker, are more concrete rather than abstract terms. Here, the high-level features (denoted as COMF tokens) of each shot are donated by IBM team [].

2.2 Parameter Setup

If a reporter mentions some specific terms, which may occur in the following shots and be annotated in the high-level annotations, there exist links between the ASR tokens and the COMF tokens. In other words, the specific term in the former mentioned ASR tokens actually "belong to" the later shots that have COMF annotations related to that specific term. This phenomenon usually occurs within a window size.

We adopt a distance metric to measure the similarity of an ASR token and a COMP token. Figure 2 shows the distance between the *first sense* of *outdoors* and the *first sense* of *tom*. Each node represents a synset entry in WordNet, and the distance between the two senses s_i and s_j, denoted $dist(s_i, s_j)$, is computed as the sum of lengths of the two paths starting from a common ancestor. In this example, the distance is 6.

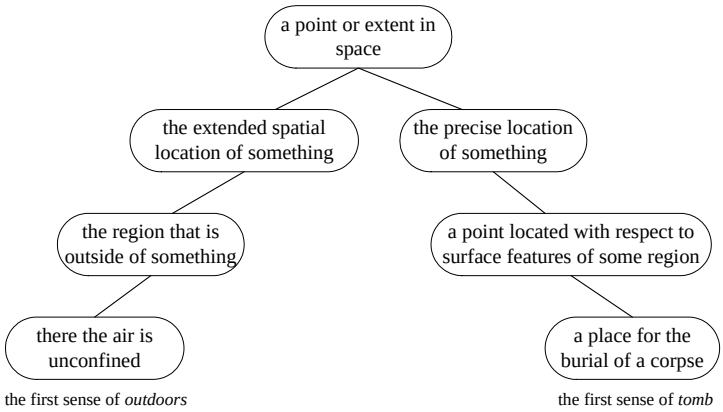


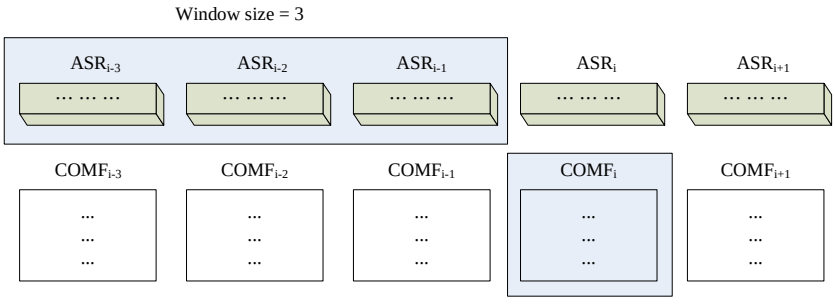
Figure 2. An Example of Distance Measurement

Because an ASR token may have multiple senses, the distance of two words w_i and w_j can be defined as the minimal value of all possible pairs of sense distances shown as follows.

$$dist(w_i, w_j) = \min_{si \in wi, sj \in wj} (dist(s_i, s_j)) \quad (1)$$

Where s_i and s_j denotes possible senses of word w_i and w_j, respectively.

Locality issue is also considered during finding the co-reference relationship between ASR and COMF tokens. We postulate that the relationship cannot be very far away, so that a pre-defined window size is set. Only the candidates within a window are considered. Figure 3 illustrates an example. Here the window size is set to three, meaning that we look forward three shots to see if there are ASR tokens that match the content (i.e., COMP tokens) of this shot. A *threshold*, which is also a pre-defined parameter, determines the minimum degree of similarity. Lower threshold means that the two words w_i and w_j must be close enough for w_j to be aligned. In other words, lower value means more strict conditions on word likelihood estimation.



For each w_i in the high-level feature annotations
  For each w_j in ASR within a fixed window size
    if(dist(w_i, w_j) < threshold) then
      add w_j to the doc_{shoti}
    else {}

Figure 3. An Illustration of Alignment Algorithm

Each shot is represented as a document shown in Figure 4. The first and the second parts are the original ASR and the original high-level feature annotations of this shot, respectively. The third part is the ASR tokens selected from the previous shots within a window size using the alignment algorithm. The representative document is used for indexing.

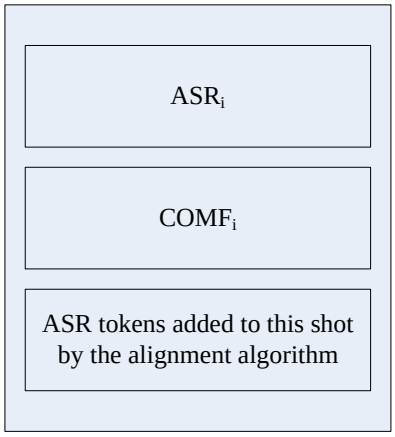


Figure 4. Representative Document Structure of $Shot_i$

To examine the effects of different parameter setting, we use the TRECVID 2003 test dataset, which contains 131 mpeg files, with the ASR outputs and the high-level feature annotations. Okapi is employed as our information retrieval system. First we fixed window size to be 4, and the sense number to be 1 (i.e., we consider only the first sense of each word), and test different threshold settings. The result is shown in Table 1. ASR denoted the baseline IR performance when only ASR is considered as a representative document; w4s1d2 denotes that the window size is set to 4, sense number set to 1, and the threshold set to 2, and so on. The average precision and the R-precision is measured using the TRECVID-supplied evaluation tool. It shows that both average precision and R-precision are increased when the alignment algorithm is performed.

Table 1. Performance Comparison with Different Thresholds

	ASR	**w4s1d2**	**w4s1d4**	**w4s1d8**
AvP	0.0413	0.0436	0.0450	0.0465
R-P	0.0649	0.0679	0.0688	0.0663

We also evaluated IR results with different window size and different sense numbers. The results with different window sizes are listed in Table 2. When the window size increases, the information brought in from ASR results by alignment increases, so that the IR performance increases. However, when the window size is increased to 8, noises along with information come in. That results a worse alignment and the performance drops.

The test result with different sense numbers is shown in Table 3. In WordNet, senses are arranged according to their frequencies of senses. Here the parameter *sn* denotes that the senses of the *first n* higher frequencies are used. Symbol *sa* denotes all the senses of a token are explored. The results with different window size and different word-to-word threshold settings show the same trend.

Table 2. Performance Comparison with Different Window Sizes

	ASR	w2s1d8	w4s1d8	w6s1d8	w8s1d8
AvP	0.0413	0.0463	0.0465	0.0470	0.0441
R-P	0.0649	0.0671	0.0663	0.0660	0.0641

Table 1 and Table 3 conclude that when the window size is fixed, choosing the first (the most frequently used) sense and looser condition for computing word-to-word distance achieves better performance. This is because we should set a broader range so that the related words can be aligned correctly, when the most likely senses are used. Table 3 shows that noises are introduced from the wrong senses when the first three senses are taken into consideration. In the extreme cases, i.e., all the senses of words are considered, much more noises from irrelevant (even rarely used) senses are added and thus the performance drops further. With the larger sense numbers and the narrower word-to-word distance threshold, our preliminary experiments show that the unrelated words are added to the representative document. For example, the word *music* has a synset entry whose meaning is "punishment for one's actions", passing the threshold setting, and is aligned with the word *responsibility*. Although the two senses are similar enough, the sense of *music* mentioned above is rarely used and should not be used for measuring the two words' distance.

Table 3. Performance Comparison with Different Sense Numbers

	ASR	w4s1d8	w4s3d8	w4sad8
AvP	0.0413	0.0465	0.0459	0.0453
R-P	0.0649	0.0663	0.0673	0.0687

Below show some examples to demonstrate the effects of our alignment algorithm and its improvement to video IR. Consider the TRECVID 2003 topic *0105: Find shots of a helicopter in flight or on the ground*. Our system successfully aligned the word *helicopter* to the right shot *shot193_41* by using the high-level feature *Airplane* in this shot. Apparently the word *helicopter* will not appear in the representative document of *shot193_41* if we do not apply our algorithm. Because the word *helicopter* was correctly aligned and added to the corresponding representative document, our system can retrieve those shots that were not retrieved by traditional text retrieval using ASR only. Consider the topic *0106: Find shots of the Tomb of the Unknown Soldier at Arlington National Cemetery*. Our system successfully aligned the word *tomb* to *shot146_65* and *cemetery* to *shot178_176* by the high-level feature *Outdoors*, as illustrated in Figure 1. Using ASR only failed again.

We also discuss in which query types our system performs better. We classify the queries to three categories – say, *Object*, *Scene*, and *Person*, and find that our system performs better in some queries while worse in other queries. This might be a *generic* characteristic of our algorithm, that is, we do not build different modules (e.g., face detector and so on) to deal with different queries. Instead we use a general and an efficient way to deal with different kinds of queries.

Our algorithm is simple and fast, and it does improve the video IR performance by aligning the ASR tokens to its right shots. However, without using low-level video content information, our algorithm fails to build the correct alignment under some conditions. For example, if the topic is to find the *shots with an airplane*

taking off, a relevant shot may be an event to rescue some person or to airdrop in some area. The word *airplane* will not appear in the ASR, thus we cannot align *airplane* to this shot because it does not appear. Unless the high-level feature annotated the word *airplane*, this shot seems to be irrelevant by the IR system. In other words, if the focus is on a specific *event* rather than the *object* or *person* we see on the screen, our system would not be able to build the right connections between the ASR tokens and the COMF tokens. Besides, our system is sensitive to the high-level features. In the experiments with TRECVID 2003 data, the number of high-level features seems to be small, and most of the shots contain only zero or one annotation. We believe that using a larger number of high-level features will further increase the correct alignments and the IR performance of our system.

3 Experimental Results

In the TRECVID 2004 search task we choose parameter w4s1d8 and w6s1d8, using the Okapi as our IR system, retrieve relevant shots, and then return them to NIST for assessment. The results are shown in Table 4 and Figure 5. System/run IDs *M_C_1_NLP_Lab1_1*, *M_C_1_NLP_Lab1_1***,** and *M_C_1_NLP_Lab1_1* denote the ASR baseline, the results with parameter w4s1d8, and results with parameter w6s1d8, respectively.

Table 4. Performance on TRECVID 2004 Search Task

	M_C_1_NLP_Lab1_1	**M_C_2_NLP_Lab1_1**	**M_C_2_NLP_Lab1_1**
AvP	0.024	0.026	0.026
R-P	0.099	0.105	0.100

The experimental results show that the baseline ASR does not perform well, and the improvement of the alignment algorithm seems to be limited. One of the reasons for the low performance in the baseline model is that the task stresses on the importance of the video content rather than news script content, and the keywords in the chosen topics usually do not appear in the ASR words. Furthermore, this characteristic affects our algorithm in aligning the words to their shots. As illustrated in the last section, if the story in the news event does not focus on an *object* we would like to search for, we could not find the related words around this shot.

Although the overall improvement seems limited in this task, Figure 5 shows that our alignment algorithm helps in certain topics. For example, the average precision of topic 137 increases from 8.7% to 11.2%, and that of topic 134 increases from17.1% to 31.5%. However, the alignment algorithm helps little in more topics. That results in the low performance improvement in total. The major reason is the annotation errors from the automatic generated high-level features, and the propagation errors bring more noises.

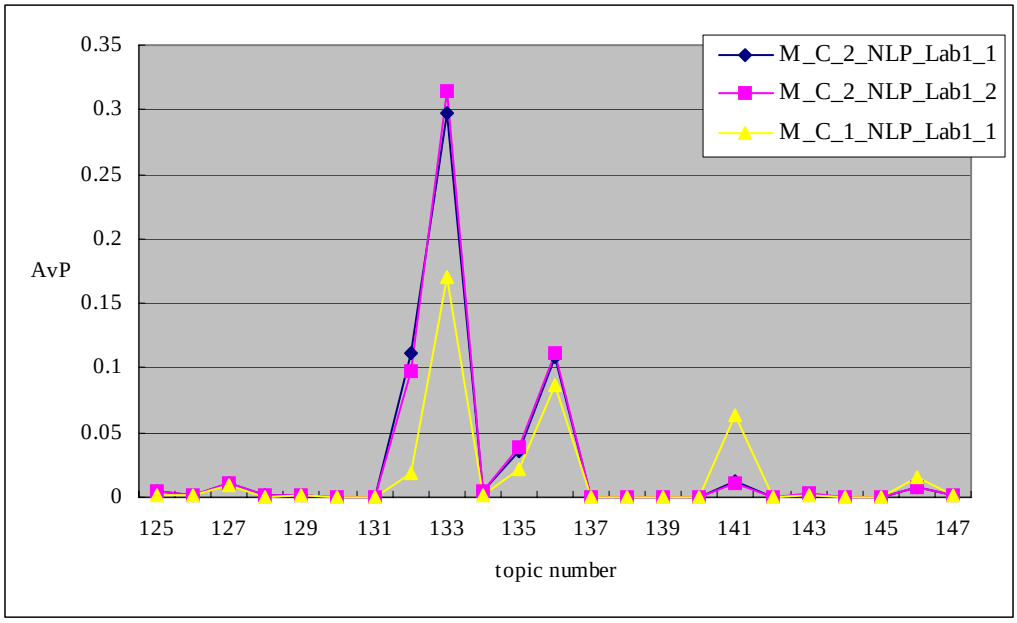


Figure 5. Performance on TRECVID 2004 Search Task

4 Conclusions and Future Work

In summary, this paper proposed an approach to align the ASR word tokens to the corresponding shots by calculating their word-to-word distance with the high-level feature COMF tokens. Without complex algorithms and plenty of computing time, this method does lead to an improvement of the performance of the video information retrieval. We apply this approach on the TRECVID 2003 test data, test the parameters of our system, and then use these parameters to run with the TRECVID 2004 test data. In some topics the alignment algorithm improves the performance significantly, however, in other topics not. By low computing cost and simple algorithm, our method can increase the overall video IR performance. We believe that this algorithm has a potential to improve the video IR performance further. In the future we will investigate methods for automatically generate the high-level features, because our algorithm is sensitive to these features. And we will continue on extending our research on different corpus, and specifying or modifying our algorithm to deal with different query types.

References

1. Hauptmann, A., Ng, T.D., and Jin, R. "Video Retrieval using Speech and Image Information," *Proceedings of 2003 Electronic Imaging Conference*, Storage and Retrieval for Multimedia Databases, Santa Clara, CA, January 20-24, 2003
2. Duygulu, P., and Wactlar, H. "Associating Video frames with Text," *Proceedings of the 26th Annual International ACM SIGIR Conference*, Toronto, Canada, July 28-August 1, 2003
3. Hauptmann, A., Chen, M-Y., and Duygulu, P. "What's News, What's Not? Associating News Videos with Words," *Proceedings of the 3rd International Conference on Image and Video Retrieval*, Dublin City University, Ireland, July 21-23, 2004
4. E. Brill. "A Simple Rule-based Part of Speech Tagger," *Proceedings of the Third Conference on Applied Natural Language Processing*, 1992.