Toward Multimedia: A String Pattern-based Passage Ranking Model for Video Question Answering

Yu-Chieh Wu
Dept. of Computer Science and Information Engineering
National Central University
Taoyuan, Taiwan
bcbb@db.csie.ncu.edu.tw

Jie-Chi Yang
Graduate Institute of Network Learning Technology
National Central University
Taoyuan, Taiwan
yang@cl.ncu.edu.tw

Abstract

In this paper, we present a new string pattern matching-based passage ranking algorithm for extending traditional text-based QA toward videoQA. Users interact with our videoQA system through natural language questions, while our system returns passage fragments with corresponding video clips as answers. We collect 75.6 hours videos and 253 Chinese questions for evaluation. The experimental results showed that our method outperformed six top-performed ranking models. It is 10.16% better than the second best method (language model) in relatively MRR score and 6.12% in precision rate. Besides, we also show that the use of a trained Chinese word segmentation tool did decrease the overall videoQA performance where most ranking algorithms dropped at least 10% in relatively MRR, precision, and answer pattern recall rates.

1 Introduction

With the drastic growth of video sources, effective indexing and retrieving video contents has recently been addressed. The well-known Informedia project (Wactlar, 2000) and TREC-VID track (Over et al., 2005) are the two famous examples. Although text-based question answering (QA) has become a key research issue in past decade, to support multimedia such as video, it is still beginning.

Over the past five years, several video QA studies had investigated. Lin et al. (2001) presented an earlier work on combining videoOCR and term weighting models. Yang et al. (2003) proposed a complex videoQA approach by employing abundant external knowledge such as, Web, WordNet, shallow parsers, named entity taggers, and human-made rules. They adopted the term-weighting method (Pasca, and Harabagiu, 2001) to rank the video segments by weighting the pre-defined keywords. Cao and Nunamaker (2004) developed a lexical pattern matching-based ranking method for a domain-specific videoQA. In the same year, Wu et al. (2004) designed a cross-language (English-to-Chinese) video question answering system based on extracting pre-defined named entity words in captions. On the other hand, Zhang and Nunamaker (2004) made use of the simple TFIDF term weighting schema to retrieve the manual-segmented clips for video caption word retrieval. They also manually developed the ontology to improve system performance.

In this paper, we present a new string pattern matching-based passage ranking algorithm for video question answering. We consider that the passage is able to answer questions and also suitable for videos because itself forms a very natural unit. Lin et al. (2003) showed that users prefer passage-level answers over short answer phrases since it contains rich context information. Our method makes use of the string pattern searching in the suffix trees to find common subsequences between a passage and question. The proposed term weighting schema is then designed to compute passage score. In addition, to avoid generating over-length subsequence, we also present two algorithms for re-tokenization and weighting.

2 The Framework of our VideoQA System

An overview of the proposed videoQA system can be shown in Figure 1. The video processing component recognizes the input video as an OCR document at the first stage. Second, each three consecutive sentences were grouped into a passage. We tokenized the Chinese words with three grained sizes: unigram, bigram, and trigram. Similarly, the input question is also tokenized to uni-

Proceedings of NAACL HLT 2007, pages 540–547,
Rochester, NY, April 2007.

gram, bigram, and trigram level of words. To reduce most irrelevant passages, we adopted the BM-25 ranking model (Robertson et al., 2000) to retrieve top-1000 passages as the “input passages”. Finally, the proposed passage ranking algorithm retrieved top-*N* passages as answers in response to the question. In the following parts, we briefly introduce the employed videoOCR approach. Section 2.2 presents the sentence and passage segmentation schemes. The proposed ranking algorithms will be described in Section 3.

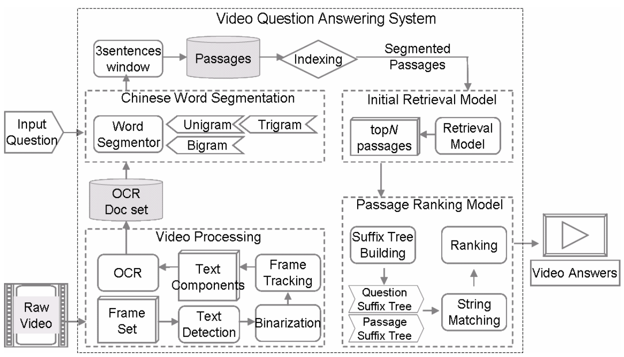

Figure1: System Architecture of the proposed videoQA system

2.1 Video Processing

Our video processing takes a video and recognizes the closed captions as texts. An example of the input and output associated with the whole video processing component can be seen in Figure 2. The videoOCR technique consists of four important steps: text detection, binarization, frame tracking, and OCR. The goal of text detection is to locate the text area precisely. In this paper, we employ the edge-based filtering (Lyu et al., 2005) and slightly modify the coarse-to-fine top-down block segmentation methods (Lienhart and Wernicke, 2002) to find each text component in a frame. The former removes most non-edge areas with global and local thresholding strategy (Fan et al., 2001) while the latter incrementally segments and refines text blocks using horizontal and vertical projection profiles.

The next steps are text binarization and frame tracking. As we know, the main constituent of video is a sequence of image frames. A text component almost appears more than once. To remove redundancy, we count the proportion of overlapping edge pixels between two consecutive frames. If the portion is above 70%, then the two frames were considered as containing the same text components. We then merge the two frames by averaging the gray-intensity for each pixel in the same text component. For the binarization stage, we employ the Lyu’s text extraction algorithm (Lyu et al., 2005) to binarize text pixels for the text components. Unlike previous approaches (Lin et al., 2001; Chang et al., 2005), this method does not need to assume the text is in either bright or dark color (but assume the text color is stable). At the end of this step, the output text components are prepared for OCR.

The target of OCR is to identify the binarized text image to the ASCII text. In this paper, we developed a naïve OCR system based on nearest neighbor classification algorithms and clustering techniques (Chang et al., 2005). We also adopted the word re-ranking methods (Lin et al., 2001, strategy 3) to improve the OCR errors.

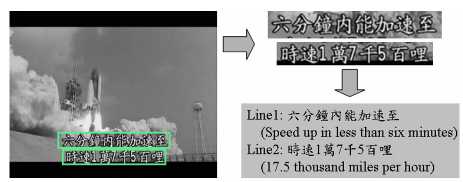

Figure 2: Text extraction results of an input image

2.2 Sentence and Passage Segmentation

In this paper, we treat all words appear in the same frame as a sentence and group every three consecutive sentences as a passage. Usually, words that occur in the same frame provide a sufficient and complete description. We thus consider these words as a sentence unit for sentence segmentation. An example of a sentence can be found in Figure 2. The sentence of this frame is the cascading of the two text lines, i.e. “speed-up to 17.5 thousand miles per hour in less than six minutes” For each OCR document we grouped every three continuous sentences with one previous sentence overlapping to represent a passage. Subsequently, we tokenized Chinese word with unigram, bigram, and trigram levels.

Searching answers in the whole video collection is impractical since most of them are irrelevant to the question. By means of text retrieval technology, the search space can be largely reduced and limited in a small set of relevant document. The document retrieval methods have been developed well and successfully been applied for retrieving relevant passages for question answering (Tellex et al., 2003). We replicated the Okapi BM-25 (Robertson et al., 2000), which is the effective and efficient retrieval algorithms to find the related segmented passages. For each input question, the top-1000 relevant passages are input to our ranking model.

3 The Algorithm

Tellex et al. (2003) compared seven passage retrieval models for text QA except for several adhoc approaches that needed either human-generated patterns or inference ontology which were not available. In their experiments, they showed that the density-based methods (Lee et al., 2001) achieved the best results, while the BM-25 (Robertson, 2000) reached slightly worse retrieval result than the density-based approaches, which adopted named entity taggers, thesaurus, and WordNet. Cui et al. (2005) showed that their fuzzy relation syntactic matching method outperformed the density-based methods. But the limitation is that it required a dependency parser, thesaurus, and training data. In many Asian languages like Chinese, Japanese, parsing is more difficult since it is necessary to resolve the word segmentation problem before part-of-speech (POS) tagging, and parsing (Fung et al., 2004). This does not only make the parsing task harder but also required to train a high-performance word segmentor. The situation is even worse when text contains a number of OCR error words. In addition, to develop a thesaurus and labeled training set for QA is far time-consuming. In comparison to Cui's method, the term weighting-based retrieval models are much less cost, portable and more practical. Furthermore, the OCR document is not like traditional text articles that have been human-typed well where some words were error predicted, unrecognizable, and false-alarm. These unexpected words deeply affect the performance of Chinese word segmentation, and further for parsing. In our experiments (see Table 2 and Table 3), we also showed that the use of a well-trained high-performance Chinese word segmentation tool gave the worse result than using the unigram-level of Chinese word (13.95% and 13.92% relative precision and recall rates dropped for language model method).

To alleviate this problem, we treat the atomic Chinese unigram as word and present a weighted string pattern matching algorithm. Our solution is to integrate the suffix tree for finding, and encoding important subsequence information in trees. Nevertheless, it is known that the suffix tree construction and pattern searching can be accomplished in linear time (Ukkonen, 1995). Before introducing our method, we give the following notations.

passage P = $PW_1, PW_2, \ldots, PW_T$

question Q = $QW_1, QW_2, \ldots, QW_{T'}$

a common subsequence for passage

$$\mathrm{Sub}_i^{\mathrm{P}} = \mathrm{PW}_k, \mathrm{PW}_{k+1}, \ldots, \mathrm{PW}_{k+x-1} \quad \text{if } |\mathrm{Sub}_i^{\mathrm{P}}| = x$$

a common subsequence for question

$$\mathrm{Sub}_j^{\mathrm{Q}} = \mathrm{QW}_l, \mathrm{QW}_{l+1}, \ldots, \mathrm{QW}_{l+y-1} \text{ if } |\mathrm{Sub}_j^{\mathrm{Q}}| = y$$

A common subsequence represents a continuous string matching between P and Q. We further impose two symbols on a subsequence. For example, $\mathrm{Sub}_i^{\mathrm{P}}$ means *i*-th matched continuous string (common subsequence) in the passage, while $\mathrm{Sub}_j^{\mathrm{Q}}$ indicates the *j*-th matched continuous string in the question. The common subsequences can be extracted through the suffix tree building and pattern searching. For example, to extract the set of $\mathrm{Sub}_i^{\mathrm{P}}$, we firstly build the suffix tree of P and incrementally insert substring of Q and label the matched common string between P and Q. Similarly, one can apply a similar approach to generate the set of $\mathrm{Sub}_j^{\mathrm{Q}}$. By extracting all subsequences for P and Q, we then compute the following score (see equation (1)) to rank passages.

$$\begin{aligned}&\mathrm{Passage\_Score(P)} = \lambda \times \mathrm{QW\_Density(Q,P)} + \\ &(1-\lambda) \times \mathrm{QW\_Weight(Q,P)}\end{aligned} \tag{1}$$

The first term of equation (1) "QW_Density(Q, P)" estimates the question word density degree in the passage P, while "QW_Weight(Q, P)" measures the matched question word weights in P. λ is a parameter, which is used to adjust the importance of the QW_Density(Q, P). Both the two estimations make use of the subsequence information for P and Q. In the following parts, we introduce the computation of QW_Density(Q,P) and QW_Weight(Q, P) separately. The time complexity analysis of our method is then discussed in the tail of this section.

The QW_Density(Q, P) is designed for quantifying "how dense the matched question words in the passage P". It also takes the term weight into account. By means of extracting common subsequence in the question, the set of $\mathrm{Sub}_j^{\mathrm{Q}}$ can be used to measures the question word density. At the beginning, we define equation (2) for weighting a subsequence $\mathrm{Sub}_j^{\mathrm{Q}}$.

$$\text{Weight}(\text{Sub}_j^Q) = \text{length}(\text{Sub}_j^Q)^{\alpha_1} \times \text{DP}(\text{Sub}_j^Q) \quad (2)$$

Where length(Sub_j^Q) is merely the length of Sub_j^Q i.e., the number of words in Sub_j^Q. α_1 is a parameter that controls the weight of length for Sub_j^Q. In this paper, we consider the long subsequence match is useful. A long *N*-gram is usually much less ambiguous than its individual unigram. The second term in equation (2) estimates the "discriminative power" (DP) of the subsequence. Some high-frequent and common words should be given less weight. To measure the DP score, we extend the BM-25 (Robertson et al., 2000) term weighting schema. Equation (3), (4), and (5) list our DP scoring functions.

$$\text{DP}(\text{Sub}_j^Q) = W' \times \frac{(k_1+1)\times \text{TF}(\text{Sub}_j^Q, \text{P})}{K + \text{TF}(\text{Sub}_j^Q, \text{P})} \times \frac{(k_3+1)\times \text{TF}(\text{Sub}_j^Q, \text{Q})}{k_3 + \text{TF}(\text{Sub}_j^Q, \text{Q})} \quad (3)$$

$$W' = \log\left(\frac{N_P - \text{PF}(\text{Sub}_j^Q) + 0.5}{\text{PF}(\text{Sub}_j^Q) + 0.5}\right) \quad (4)$$

$$K = (1-b) + b \times \frac{|\text{P}|}{\text{AVG}(|\text{P}|)} \quad (5)$$

k_1, b, k_3 are constants, which empirically set as 1.2, 0.75, 500 respectively (Robertson et al., 2000). $\text{TF}(\text{Sub}_j^Q, \text{Q})$ and $\text{TF}(\text{Sub}_j^Q, \text{P})$ represent the term frequency of Sub_j^Q in question Q and passage P. Equation (4) computes the inverse "passage frequency" (PF) of Sub_j^Q as against to the traditional inverse "document frequency" (DF) where N_p is the total number of passages. The collected Discovery video is a small but "long" OCR document set, which results the estimation of DF value unreliable. On the contrary, a passage is more coherent than a long document, thus we replace the DF estimation with PF score. It is worth to note that some Sub_j^Q might be too long to be further re-tokenized into finer grained size. We therefore propose two algorithms to 1): re-tokenize an input subsequence, and 2): compute the DP score for a subsequence. Figure 3, and Figure 4 list the proposed two algorithms.

The proposed algorithm 1, and 2 can be used to compute and tokenize the DP score of not only Sub_j^Q for question but also Sub_j^P for passage. As seeing in Figure 4, it requires DP information for different length of *N*-gram. As noted in Section 2.2, the unigram, bigram, and trigram level of words had been stored in indexed files for efficient retrieving and computing DP score at this step. By applying algorithm 1 for the set of Sub_j^Q, we can obtain all retokenized subsequences (TSub_j). We then use the re-tokenized subsequences to compute the final density score. Equation (6) lists the QW_Density scoring function.

$$\text{QW\_Density}(\text{Q},\text{P}) = \sum_{i=1}^{T\_CNT-1} \frac{\text{Weight}(\text{TSub}_i) + \text{Weight}(\text{TSub}_{i+1})}{\text{dist}(\text{TSub}_i, \text{TSub}_{i+1})^{\alpha_2}} \quad (6)$$

$$\text{dist}(\text{TSub}_i, \text{TSub}_{i+1}) = \text{min\_distance\_between}(\text{TSub}_i, \text{TSub}_{i+1})\text{\_in\_P} + 1 \quad (7)$$

T_CNT is the total number of retokenized subsequences in Q, which can be extracted through applying algorithm 1 for all Sub_j^Q. Equation (7) merely counts the minimum number of words between two neighboring TSub_i, and TSub_{i+1} in the passage. α_2 is the parameter that controls the impact of distance measurement.

```
Algorithm 1: Retokenizing_a_subsequence
Input:
    A subsequence Sub_j^Q where start_j is the position of first word in
    question and end_j is the position of last word in question
Output:
    A set of retokenized subsequence {TSub_1, TSub_2,.....}
    N_t: the number of retokenized subsequence
Algorithm:
    Initially, we set N_t := 1; TSub_1:=QW_start_j;
    if (Sub_j^Q≠ψ)
    {    /*** from the start to the end positions in the string ***/
         for ( k := start_j+1 to end_j)
         {
  /***Check the two question words is bigram in the passage***/
         if (bigram(QW_k-1,QW_k) is_found_in_passage)
              add QW_k into TSub_Nt;
         Otherwise
         {    N_t ++;
              TSub_Nt := QW_k;
         } /*** End otherwise***/
         } /*** End for ***/
    } /*** End if ***/
    else
         N_t := 0;
```

Figure 3: An algorithm for retokenizing subsequence

```
Algorithm 2: Copmuting_DP_score
Input:
    A subsequence  Sub_j^Q where start_j is the position of first word
    of Sub_j^Q in question end_j is the position of last word of Sub_j^Q in
    question
Output:
    The score of DP(Sub_j^Q)
Algorithm:
    head := start_j;
    tail := end_j;
    Max_score := 0;
    for (k := head ~ tail)
    {    let WORD := QW_k, QW_k+1,…, QW_tail;
         /*** look-up WORD in the index files  ***/
         compute DP(WORD) using equation (3);
         if (DP(WORD) > Max_score)
              Max_score := DP(WORD);
    } /*** End for ***/
    DP(WORD) := Max_score;
```

Figure 4: An algorithm for computing DP score for a subsequence

The density scoring can be thought as measuring "how much information the passage preserves in response to the question". On the contrary, the QW_Weight (second term in equation (1)) aims to estimate "how much content information the passage has given the question". To achieve this, we further take the other extracted common subsequences, i.e., $\mathrm{Sub}_j^{\mathrm{P}}$ into account. By means of the same term weighting schema for the set of $\mathrm{Sub}_j^{\mathrm{P}}$, the QW_Weight is then produced. Equation (8) gives the overall QW_Weight measurement.

$$\mathrm{QW\_Weight}(\mathrm{Q},\mathrm{P}) = \sum_{i=1}^{S\_CNT} \mathrm{Weight}(\mathrm{Sub}_i^{\mathrm{P}}) = \sum_{i=1}^{S\_CNT} (\mathrm{length}(\mathrm{Sub}_i^{\mathrm{P}})^{\alpha_1} \times \mathrm{DP}(\mathrm{Sub}_i^{\mathrm{P}})) \tag{8}$$

where the DP score of the input subsequence can be obtained via the algorithm 2 (Figure 5). *S_CNT* is the number of subsequence in P. The parameter α_1 is also set as equal as equation (2).

In addition, the neighboring contexts of a sentence, which contains high QW_Density score might include the answers. Hence, we stress on either head or tail fragments of the passage. In other words, the passage score is determined by computing equation (1) for head and tail parts of passage. We thus extend equation (1) as follows.

$$\begin{aligned}\mathrm{Passage\_Score}(\mathrm{P}) = \max\{ & \lambda \times \mathrm{QW\_Density}(\mathrm{Q},\mathrm{P}_1) + (1-\lambda) \times \mathrm{QW\_Weight}(\mathrm{Q},\mathrm{P}_1), \\ & \lambda \times \mathrm{QW\_Density}(\mathrm{Q},\mathrm{P}_2) + (1-\lambda) \times \mathrm{QW\_Weight}(\mathrm{Q},\mathrm{P}_2)\}\end{aligned}$$

$$\begin{cases} \text{if P has 3 sentences}: & \mathrm{S}_1,\mathrm{S}_2,\mathrm{S}_3 & \text{then}, \mathrm{P}_1 = \mathrm{S}_1 + \mathrm{S}_2 \text{ and } \mathrm{P}_2 = \mathrm{S}_2 + \mathrm{S}_3 \\ \text{else if P has 2 sentences}: & \mathrm{S}_1,\mathrm{S}_2 & \text{then}, \mathrm{P}_1 = \mathrm{S}_1 \text{ and } \mathrm{P}_2 = \mathrm{S}_2 \\ \text{else if P has 1 sentence}: & \mathrm{S}_1 & \text{then}, \mathrm{P}_1 = \mathrm{P}_2 = \mathrm{S}_1 \end{cases}$$

Instead of estimating the whole passage, the two divided parts: P_1, and P_2 are used. We select the maximum passage score from either head (P_1) or tail (P_2) part. When the passage contains only one sentence, then this sentence is indispensable to be used for estimation.

Now we turn to analyze the time complexity of our algorithm. It is known that the suffix tree construction costs is linear time (assume it requires $O(T)$, T: the passage length for passage and $O(T')$, T': the question length for question). Assume the search time for a pattern in the suffix trees is at most $O(h\log m)$ where h is the tree height, and m is the number of branch nodes. To generate the sets of $\mathrm{Sub}_j^{\mathrm{Q}}$ and $\mathrm{Sub}_j^{\mathrm{P}}$, it involves in building suffix trees and incrementally searching substrings, i.e., $O((T+T')+(T+T')(h\log m))$. Intuitively, both algorithm 1, and algorithm 2 are linear time algorithms, which depends on the length of "common" subsequence, i.e., at most $O(\min(T, T'))$. Consequently, the overall time complexity of our method for computing a passage is $O((T+T')(1+h\log m)+\min(T, T'))$.

4 Experiments

4.1 Evaluation

We should carefully select the use of videoQA collection for evaluation. Unfortunately, there is no benchmark corpus for this task. Thus, we develop an annotated collection by following the similar tasks as TREC, CLEF, and NTCIR. The Discovery videos are one of the popular raw video sources and widely evaluated in many literatures (Lin et al., 2001; Wu et al., 2004; Lee et al., 2005). Totally, 75.6 hours of Discovery videos (93 video names) were used. Table 1 lists the statistics of the Discovery films.

The questions were created in two different ways: one set (about 73) was collected from previous studies (Lin et al., 2001; Wu et al., 2004) which came from the "Project: Assignment of Discovery"; while the other was derived from a real log from users. Video collections are difficult to be general-purpose since hundreds hours of videos might take tens of hundreds GB storage space. Therefore, general questions are quite difficult to be found in the video database. Hence, we provide a list of short introductions collected from the cover-page of the videos and enable users to browse the descriptions. Users were then asked for the system with limited to the collected video topics. We finally filter the (1) keyword-like queries (2) non-Chinese and (3) un-supported questions. Finally, there were 253 questions for evaluation.

For the answer assessment, we followed the TREC-QA track (Voorhees, 2001) and NTCIR to annotate answers in the pool that collected from the outputs of different passage retrieval methods. Unlike traditional text QA task, most of the OCR sentences contain a number of OCR error words. Furthermore, some sentence did include the answer string but error recognized as different words. Thus, instead of annotating the recognized transcripts, we used the corresponding video frames for evaluation because users can directly find the answers in the retrieved video clips and recognized text. Among 253 questions, 56 of which did not have an answer, while 368 passage&frame segments (i.e., answer patterns) in the pool were labeled as answers. On averagely, there are 1.45 labeled answers for each question.

The MRR (Voorhees, 2001) score, precision and pattern-recall are used for evaluation. We measure the MRR scores for both top1 and top5 ranks, and precision and pattern-recall rates for top5 retrieved answers.

Table 1: Statistics of the collected Discovery videos

# of videos	# of sentence	# of words	# of passages
93	49950	746276	25001
AVG # of words per sentence	AVG # of words per passage	AVG # of sentences per passage	AVG # of words per video
14.94	48.78	537.09	8024.47

4.2 Results

In this paper, we employed six top-performed yet portable ranking models, TFIDF, BM-25 (Robertson et al., 2000), INQUERY, language model (Zhai and Lafferty, 2001), cosine, and density-based (Lee et al., 2001) approaches for comparison[1]. For the language model, the Jelinek-Mercer smoothing method was employed with the parameter settings λ=0.5 which was selected via several trials. In our preliminary experiments, we found that the query term expansion does not improve but decrease the overall ranking performance for all the ranking models. Thus, we only compare with the "pure" retrieval performance without pseudo-feedback.

The system performance was evaluated through the returned passages. We set α_1=1.25, α_2= 0.25, and λ=0.8 which were observed via the following parameter validations. More detail parameter experiments are presented and discussed later. Table 2 lists the overall videoQA results with different ranking models.

Among all ranking models, the proposed method achieves the best system performance. Our approach produced 0.596 and 0.654 MRR scores when evaluating the top1 and top5 passages and the precision rate achieves 0.208. Compared to the second best method (language model), our method is 10.16% better in relatively percentage in terms of MRR(top1) score. For the MRR(top5) score, our method is 7.39 relative percentage better. In terms of the non-answered questions, our method also covers the most questions (253-69=184) compared to the other ranking models. Overall, the experiment shows that the proposed weighted string pattern matching algorithm outperforms the other six methods in terms of MRR, non-answered question numbers, precision and pattern recall rates.

Table 2: Overall videoQA performance with different ranking models (using unigram Chinese word)

Word-Level	MRR (Top1)	MRR (Top5)	Non-answered Questions	Precision	Pattern Recall
TFIDF	0.498	0.572	81	0.189	0.649
BM-25	0.501	0.581	78	0.186	0.638
Language Model	0.541	0.609	74	0.196	0.671
INQUERY	0.505	0.583	78	0.188	0.644
Cosine	0.418	0.489	102	0.151	0.519
Density	0.323	0.421	102	0.137	0.471
Our Method	**0.596**	**0.654**	**69**	**0.208**	**0.711**

Table 3: Overall videoQA performance with different ranking models using word segmentation tools

Word-Level	MRR (Top1)	MRR (Top5)	Non-answered Questions	Precision	Pattern Recall
TFIDF	**0.509**	**0.567**	**89**	0.145	**0.597**
BM-25	0.438	0.500	104	0.159	0.543
Language Model	0.486	0.551	89	0.172	0.589
INQUERY	0.430	0.503	97	0.164	0.562
Cosine	0.403	0.480	100	0.158	0.548
Density	0.304	0.380	125	0.133	0.451
Our Method	**0.509**	**0.561**	**89**	**0.181**	**0.608**

Next, we evaluate the performance with adopting a trained Chinese word segmentation tool instead of unigram level of word. In this paper, we employed the Chinese word segmentation tool (Wu et al., 2006) that achieved about 0.93-0.96 recall/precision rates in the SIGHAN-3 word segmentation task (Levow, 2006). Table 3 lists the overall experimental results with the adopted word segmentation tool. In comparison to unigram grained level (Table 2), it is shown that the use of word segmentation tool does not improve the videoQA result for most top-performed ranking models, BM-25, language model, INQUERY, and our method. For example, our method is relatively 17.92% and 16.57% worse in MRR(Top1) and MRR(Top5) scores. In terms of precision and pattern-recall rates, it drops 14.91, and 16.94 relative percentages, respectively. For the TFIDF method, the MRR score is almost the same as previous result whereas it decreased 30.34%, and 8.71% precision and pattern-recall rates. On averagely, the four models, BM-25, language model, INQUERY, and our method dropped at least relatively 10% in MRR, precision, and pattern-recall rates. In this experiment, our ranking algorithm also achieved

[1] For the TFIDF/BM-25/INQUERY/Language Model approaches were performed using the Lemur toolkit

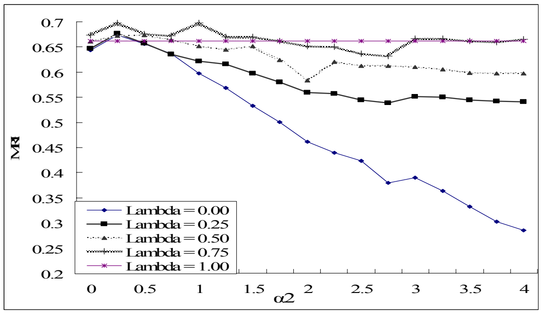

Figure 5: Experimental results with different settings of parameter α_1 using MRR evaluation

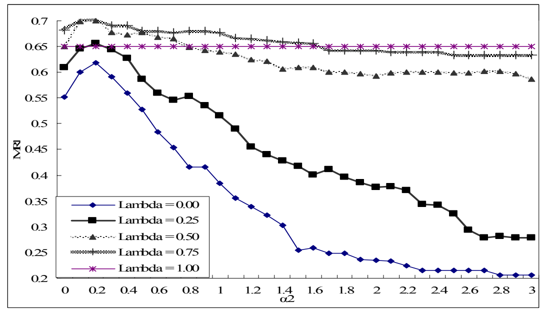

Figure 6: Verify parameter α_2 with α_1=1.25, and variant λ

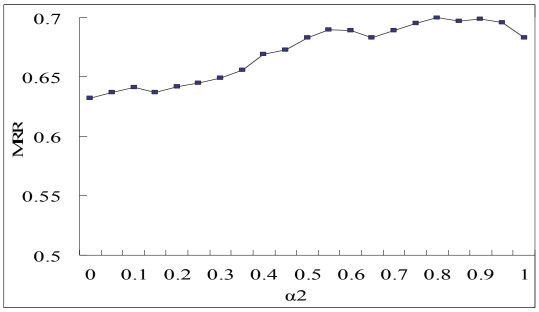

Figure 7: Verify parameter λ in the two validation sets with α_1=1.25 and α_2=0.25

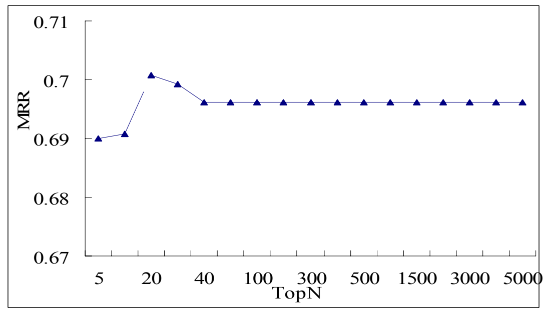

Figure 8: Experimental results with different number of initial retrieval passages (Top*N*)

the best results in terms of precision and pattern recall rates while marginally worse than the TFIDF for the MRR(top5) score.

There are three parameters: λ, α_1, α_2, in our ranking algorithm. λ controls the weight of the QW_Density(Q, P), while α_1, and α_2 were set for the power of subsequence length and the distance measurement. We randomly select 100 questions for parameter validations. Firstly, we tried to verify the optimal α_1 via different settings of the remaining two parameters. The best α_1 is then set to verify α_2 via various λ values. The optimal λ is subsequently confirmed through the observed α_1 and α_2 values. Figure 5, 6, 7 show the performance evaluations of different settings for the three parameters.

As shown in Figure 5, the optimal settings of (α_1=1.25) is obtained when and α_2=0.25, and λ=0.75. When α_1 is set more than 1.5, our method quickly decreased. In this experiment, we also found that large α_2 negatively affects the performance. The small α_2 values often lead to better ranking performance. Thus, in the next experiment, we limited the α_2 value in 0.0~3.0. As seeing in Figure 6, again the abnormal high or zero α_2 values give the poor results. This implies the over-weight and no-weight on the distance measurement (equation (7)) is not useful. Instead, a small α_2 value yields to improve the performance. In our experiment, α_2=0.25 is quite effective. Finally, in Figure 7, we can see that both taking the QW_Density, and QW_Weight into account gives better ranking result, especially QW_Density. This experiment indicates that the combination of QW_Density and QW_Weight is better than its individual term weighting strategy. When λ=0.8, the best ranking result (MRR = 0.700) is reached.

Next, we address on the impact of different number of initial retrieved passages using BM-25 ranking models. Due to the length limitation of this paper, we did not present the experiments over all the compared ranking models, while we left the further results at our web site[2]. For the three parameters, we select the optimal settings derived from previous experimental results, i.e., λ=0.8, α_1=1.25, α_2=0.25. Figure 8 shows the experimental results with different number of initial retrieved passages. When employing exactly five initial retrieved passages, it can be viewed as the re-ranking improvement over the BM-25 ranking model. As seeing in Figure 8, our method does improve the conventional BM-25 ranking approach (MRR score 0.690 v.s. 0.627) with relatively 10.04% MRR value. The best system performance is MRR=0.700 when there are merely 20 initial retrieved passages. The ranking result converges when retrieving more than 40 passages. Besides,

[2] http://140.115.112.118/bcbb/TVQS2/

we also continue the experiments using only top-20 retrieved passages on the actual 253 testing questions. The ranking performance is then further enhanced from MRR=0.654 to 0.663 with 1.37% relatively improved.

5 Conclusion

More and more users are interested in searching for answers in videos, while existing question answering systems do not support multimedia accessing. This paper presents a weighted string pattern matching-based passage ranking algorithm for extending text QA toward video question answering. We compare our method with six top-performed ranking models and show that our method outperforms the second best approach (language model) in relatively 10.16 % MRR score, and 6.12% precision rates.

In the future, we plan to integrate the other useful features in videos to support multi-model-based multimedia question answering. The video-demo version of our videoQA system can be found at the web site (http://140.115.112.118/bcbb/TVQS2/).

References

Cao, J., and Nunamaker J. F. Question answering on lecture videos: a multifaceted approach, International Conference on Digital Libraries, pages 214 – 215, 2004.

Chang, F., Chen, G. C., Lin, C. C., and Lin, W. H. Caption analysis and recognition for building video indexing systems. Multimedia systems, 10: 344-355, 2005.

Cui, H., Sun, R., Li, K., Kan, M., and Chua, T. Question answering passage retrieval using dependency relations. In Proceedings of the 28th ACM SIGIR Conference on Research and Development in Information Retrieval, pages 400-407, 2005.

Fan, J., Yau, D. K. Y., Elmagarmid, A. K., and Aref, W. G. Automatic image segmentation by integrating color-edge extraction and seeded region growing. IEEE Trans. On Image Processing, 10(10): 1454-1464, 2001.

Fung, P., Ngai, G., Yuan, Y., and Chen, B. A maximum entropy Chinese parser augmented by transformation-based learning. ACM Trans. Asian Language Information Processing, 3: 159-168, 2004.

Lee et al. SiteQ: Engineering high performance QA system using lexico-semantic pattern matching and shallow NLP. In Proceedings of the 10th Text Retrieval Conference, pages 437-446, 2001.

Lee, Y. S., Wu, Y. C., and Chang, C. H. Integrating Web information to generate Chinese video summaries. In Proceedings of 17th international conference on software engineering and knowledge engineering (SEKE), pages 514-519, 2005.

Levow, G. A. The third international Chinese language processing Bakeoff: word segmentation and named entity recognition, In Proceedings of the 5th SIGHAN Workshop on Chinese Language Processing, pages 108-117, 2006.

Lin, C. J., Liu, C. C., and Chen, H. H. A simple method for Chinese videoOCR and its application to question answering. Journal of Computational linguistics and Chinese language processing, 6: 11-30, 2001.

Lin, J., Quan, D., Sinha, V., Bakshi, K., Huynh, D., Katz, B., and Karger, D. R. What makes a good answer? the role of context in question answering. In Proceedings of the 9th international conference on human-computer interaction (INTERACT), pages 25-32, 2003.

Lienhart, R. and Wernicke, A. Localizing and segmenting text in images and videos. IEEE Trans. Circuits and Systems for Video Technology, 12(4): 243-255, 2002.

Lyu, M. R., Song, J., and Cai, M. A comprehensive method for multilingual video text detection, localization, and extraction. IEEE Trans. Circuits and Systems for Video Technology, 15(2): 243-255, 2005.

Over, P., Ianeva, T., Kraaij, W., and Smeaton, A. F. TRECVID 2005 - an overview. In Proceedings of the 14th text retrieval conference (TREC), 2005.

Pasca, M., and Harabagiu, S. High-performance question answering. In Proceedings of the 24th ACM SIGIR Conference on Research and Development in Information Retrieval, pages 366-374, 2001.

Robertson, E., Walker, S., and Beaulieu, M. Experimentation as a way of life: Okapi at TREC. Journal of Information processing and management, 36: 95-108, 2000.

Tellex, S., Katz, B., Lin, J. J., Fernandes, A., and Marton, G. Quantitative evaluation of passage retrieval algorithms for question answering. In Proceedings of the 26th ACM SIGIR Conference on Research and Development in Information Retrieval, pages 41-47, 2003.

Voorhees, E. M. Overview of the TREC 2001 question answering track. In Proceedings of the 10th Text Retrieval Conference , pages 42-52, 2001.

Ukkonen, E. Constructing suffix trees on-line in linear time. In Proceedings of the international federation of information processing, pages 484-492, 1995.

Wactlar, H. D. Informedia search and summarization in the video medium, In Proceedings of Imagina 2000 Conference, 2000.

Wu, Y. C., Lee, Y. S., Chang, C. H. CLVQ: Cross-language video question/answering system. In Proceedings of 6th IEEE International Symposium on Multimedia Software Engineering, pages 294-301, 2004.

Wu, Y. C., Yang, J. C., and Lin, Q. X. Description of the NCU Chinese Word Segmentation and Named Entity Recognition System for SIGHAN Bakeoff 2006. In Proceedings of the 5th SIGHAN Workshop on Chinese Language Processing, pages 209-212, 2006.

Yang, H., Chaison, L., Zhao, Y., Neo, S. Y., and Chua, T. S. VideoQA: Question answering on news video. In Proceedings of the 11th ACM International Conference on Multimedia, pages 632-641, 2003.

Zhai, C., and Lafferty, J. A study of smoothing methods for language models applied to ad hoc information retrieval, In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pages 334-342, 2001.

Zhang, D., and Nunamaker, J. A natural language approach to content-based video indexing and retrieval for interactive E-learning. IEEE Trans. on Multimedia, 6: 450-458, 2004.