BVideoQA: Online English/Chinese Bilingual Video Question Answering

Yue-Shi Lee
Department of Computer Science and Information Engineering, Ming Chuan University, Gwei Shan District, Taoyuan 333, Taiwan.

Yu-Chieh Wu
Department of Computer Science and Information Engineering, National Central University, Jhongli City, Taoyuan County 32001, Taiwan. E-mail: bb@db.csie.ncu.edu.tw

Jie-Chi Yang
Graduate Institute of Network Learning Technology, National Central University, Jhongli City, Taoyuan County 32001, Taiwan.

This article presents a bilingual video question answering (QA) system, namely BVideoQA, which allows users to retrieve Chinese videos through English or Chinese natural language questions. Our method first extracts an optimal one-to-one string pattern matching according to the proposed dense and long $N$-gram match. On the basis of the matched string patterns, it gives a passage score based on our term-weighting scheme. The main contributions of this approach to multimedia information retrieval literatures include: (a) development of a truly bilingual video QA system, (b) presentation of a robust bilingual passage retrieval algorithm to handle no-word-boundary languages such as Chinese and Japanese, (c) development of a large-scale bilingual video QA corpus for system evaluation, and (d) comparisons of seven top-performing retrieval methods under the fair conditions. The experimental studies indicate that our method is superior to other existing approaches in terms of precision and main rank reciprocal rates. When ported to English, encouraging empirical results also are obtained. Our method is very important to Asian-like languages since the development of a word tokenizer is optional.

Introduction

With the rapid expansion of video media sources such as television shows, news, and movies, there is an increasing demand for automatic retrieval and browsing of video data. Web-page search engines such as Google (YouTube) and Yahoo! that are purely text-based have been continuously adding images, videos, and speech to their repositories as well as emerging video information-retrieval techniques. The well-known Informedia project (Wactlar, 2000) and TRECVID competitions (Over, Ianeva, Kraaij, & Smeaton, 2007) are typical examples where users can retrieve from a video database with in-domain and structured queries.

Textual, visual, and audio information are even more frequently used for video content-based information retrieval. Among them, text information—in particular, closed caption—is the most convincible feature since it is not only closely related to the current scenario but also produces less recognition errors than do state-of-the-art speech and visual object recognition technologies. Most video-based question answering (QA) or information-retrieval approaches (C.J. Lin, Liu, & Chen, 2001; Wu, Lee, & Chang, 2004; Wu, Lee, Yang, & Yen, 2006; Yang, Chaison, Zhao, Neo, & Chua, 2003) to date have employed the Optical Character Recognition (OCR) techniques to extract text information. For this study, we focused on extracting answers from closed captions, which closely relates to video scenarios.

Further, we investigated different information (passage)-retrieval techniques for video QA for three reasons. First, it is quite difficult to locate exact answers in OCR-based documents since compared to traditional text articles, OCR transcripts contain considerable numbers of recognition errors. Second, almost all past videoQA systems (C.J. Lin et al., 2001; Wu et al., 2004; Wu, Lee, et al., 2006; Yang et al., 2003) have employed retrieval-based methods to discover answer passages. Third, answer passages provide richer and more complete context information than do short, phrase-type answers. J. Lin (2007; J. Lin et al., 2003) showed that a simple information-retrieval-based technique achieved...
very competitive results when compared to the methods (e.g., parsers) embedded within advanced natural language processing technology.

In this article, we propose a robust and bilingual passage-retrieval algorithm for videoQA. By bilingual, we mean that the question could be input in either of two different languages to retrieve the answers. Our method is the hybridized extension of our previous work (Wu, Lee, et al., 2006; Wu & Yang, 2007). First, the optimal string pattern finding algorithm (Wu, Lee, et al., 2006) is used to discover the optimally matched words. By means of the term-weighting schema (Wu & Yang, 2007), a score is then obtained. We consider that passage ranking very suitable for QA since it provides rich contextual information in answering questions. J. Lin et al. (2003) showed that users prefer passages over exact answer-phrases since the paragraph-sized chunks provide context. Our videoQA system was evaluated with 500 Chinese and 500 Chinese-to-English natural language questions and 130 hr of video media. We also compare our method with seven top-performing ranking models; namely Term Frequency × Inverse Document Frequency (TFIDF), Okapi BM-25 (Robertson, Walker, & Beaulieu, 2000; Savoy, 2005), language model (Ponte & Croft, 1998; Zhai & Lafferty, 2001, 2002), INQUERY (Broglia, Croft, Callan, & Nachbar, 1995; Ponte & Croft, 1998), cosine, and SiteQ (G.G. Lee et al., 2001) approaches with the same set of questions and videos.

The remainder of this article is organized as follows. We compare several related works which work with the same line, and then discuss and compare the developed video corpus with that in previous literatures. Next, we give an overview of the proposed bilingual videoQA system. Following the overview, we describe the presented passage-retrieval algorithm and then present the comprehensive videoQA system-performance evaluation in both Chinese and English. Finally, we discuss our conclusions, the study limitations, and future research.

Related Works

C.J. Lin et al. (2001) produced an earlier work on combining a simple video Optical Character Recognition (OCR) and lexical term weighting methods to address the problem of videoQA. To improve the OCR performance, they presented three strategies to reduce the errors. In 2003, Yang et al. proposed a very complex videoQA system which needed to be incorporated with syntactic parsers, named entity taggers, WordNet, the Internet, and human-made rules. Cao and Nunamaker (2004) and Cao, Roussinov, Robles, and Nunamaker (2005) adopted a pattern-matching-based method for online lecture videoQA, where the pattern set was constructed manually. In the same year, Wu et al. (2004) designed the first cross-language (English-to-Chinese) videoQA system based on weighting-trigger named entity words. The methodology enabled users to retrieve Chinese videos with English queries only. More recently, Zhang and Nunamaker (2004) made use of simple TFIDF ranking model to search the manually segmented video clips. They also combined their approaches with the WordNet thesaurus and domain ontology to include more lexical information. More recently, Wu, Lee, et al., 2006; Wu & Yang, 2007) designed various retrieval algorithms for Chinese videoQA.

Previous studies have not clearly indicated the benefit of using external resources, but give small evaluations on their videoQA systems. Given their complex approaches, it is quite difficult to see the main contributions from which component of their work. Although Wu et al. (2004) proposed an earlier cross-language videoQA system, it was cross-lingual instead of bilingual. More specifically, their approach did not support Chinese questions since most of the kernel components (i.e., named entity taggers) were based upon English. Furthermore, only a few studies (e.g., Wu, Lee, et al., 2006; Wu & Yang, 2007) have provided large-scale evaluations for their videoQA systems. For example, Cao and Nunamaker (2004), Cao et al. (2005), C.J. Lin et al. (2001), Wu et al. (2004), and Zhang & Nunamaker (2004) tested their approaches with only 30 to 40 hand-crafted questions and used only several hours in length videos. Such a small video collection contains merely tens of thousands of words, which means that the quantitative results cannot easily be projected on a large-scale dataset or reflect actual real-world cases. Moreover, they ignore the comparisons of recent top-performing retrieval algorithms, such as language models (Ponte & Croft, 1998; Zhai & Lafferty, 2001, 2002) and Okapi BM-25 (Robertson et al., 2000; Savoy, 2005).

Development of the VideoQA Collection

There are three main constituents in the videoQA testing corpus: (a) raw video data, (b) testing questions, and (c) annotated answers, and we shall carefully select the choice of testing corpus. Unfortunately, there is no benchmark suitable for this purpose. Thus, it was necessary to develop a new videoQA dataset for evaluation. The “Discovery” video corpus is one of the more popular raw video sources and has been widely evaluated in the literature (Y.S. Lee, Wu, & Chang, 2005; C.J. Lin et al., 2001; Wu et al., 2004; Wu, Lee, et al., 2006; Wu & Yang, 2007). In this study, we collected 130 hr of Discovery videos (180 video films) as raw video data. All videos were legally permitted to be displayed domestically at the National Central University under the copyright protection. Answers were restricted to those derived from the raw videos. Figure 1 summarizes the ratio of different video topics within the raw Discovery dataset. Table 1 provides the statistics relating to the collected videos.

Questions are restricted to fact-based (i.e., factoid), short-answer questions such as “What is the largest animal in the world?” The answers to these questions are often entities, or sentences that include a sufficient and explicit description as response. Finally, there were 500 questions for evaluation. Table 2 lists samples of the collected testing questions. The question set has been derived from our previous work (Wu et al., 2004; Wu, Lee, et al., 2006; Wu & Yang, 2007), which collected 47 questions from the “Project: Assignment of Discovery,” with the remaining 453 questions drawn from
FIG. 1. Video topic distributions of the collected Discovery films.

TABLE 1. Statistics for the collected Discovery videos.

<table>
<thead>
<tr>
<th>Items</th>
<th>Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of video films</td>
<td>180</td>
</tr>
<tr>
<td>No. of recognized sentences</td>
<td>87,727</td>
</tr>
<tr>
<td>No. of recognized words</td>
<td>1,295,048</td>
</tr>
<tr>
<td>and English word</td>
<td></td>
</tr>
<tr>
<td>No. of segmented passages</td>
<td>43,910</td>
</tr>
<tr>
<td>Average no. of words per video</td>
<td>7,194.71</td>
</tr>
<tr>
<td>Average no. of words per passage</td>
<td>48.09</td>
</tr>
<tr>
<td>Average no. of words per sentence</td>
<td>14.76</td>
</tr>
<tr>
<td>Average no. of sentences per video</td>
<td>487.37</td>
</tr>
<tr>
<td>Total data size</td>
<td>~100 GB</td>
</tr>
</tbody>
</table>

TABLE 2. Sample of the testing questions.

<table>
<thead>
<tr>
<th>Chinese</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>什麼是生命中必須的元素</td>
<td>Which is the necessary element in life?</td>
</tr>
<tr>
<td>太陽的主要組成物質</td>
<td>What are the main constituents of sun?</td>
</tr>
<tr>
<td>地球最大的動物是什麼</td>
<td>What is the largest animal in the world?</td>
</tr>
<tr>
<td>誰發明了煙火</td>
<td>Who invent the fireworks?</td>
</tr>
<tr>
<td>FBI的總部位在哪兒</td>
<td>Where is FBI?</td>
</tr>
<tr>
<td>誰建造了最早的金字塔</td>
<td>Who built the first Pyramid?</td>
</tr>
<tr>
<td>埃及圖書館的圖片</td>
<td>Show me the picture of Ness Monster.</td>
</tr>
<tr>
<td>冰雪如何形成</td>
<td>How does the hail evolve?</td>
</tr>
<tr>
<td>老虎和獅子誰會是戰爭的勝者</td>
<td>Who will be the winner of tigers and pythons?</td>
</tr>
<tr>
<td>森林之王是指誰</td>
<td>Who is the classic? (the king of the forest)</td>
</tr>
<tr>
<td>雅典娜女神像在哪裡</td>
<td>Where is Athena the Minstrel?</td>
</tr>
<tr>
<td>續粹的著名符號是什麼</td>
<td>What is the Nazi mark?</td>
</tr>
</tbody>
</table>

a real question log generated by users. Initially, we built a basic prototype videoQA system that enabled users to retrieve videos. The prototype system was mainly replicated from a previous study (Wu et al., 2004) without any new parameter settings. Owing to the limited transmission speed of the Internet and Intellectual Property policy, we restricted the system so that only domestic users could retrieve the video from the system. Other users could retrieve only static and protected images (derived from the corresponding video clips). All of the questions were in Chinese.

Video collections are difficult to make general purpose since a 100 hr of videos might take terabytes of storage space. Therefore, general questions are quite difficult to find answers for within the video database. Hence, we provided a list of short introductions collected from the cover page of the videos and allowed users to browse the descriptions. Users could then query our system with questions about similar video topics and domains. We finally eliminated (a) keyword-like queries, (b) non-Chinese queries, and (c) out-of-domain queries (i.e., those not covered within the collected Discovery video topics). We also removed the nonfactoid-type questions.

For the answer assessment, we adopted a pool-based approach which was widely employed in TREC-QA track (Dang, Kelly, & Lin, 2007; Voorhees, 1999, 2000, 2001). The pool first collects candidate answers from different passage-retrieval algorithms. Then, a native-language speaking expert read each candidate answer and made a binary decision to determine whether the candidate actually did contain an answer (positive related) to the question in the context of the passage. For example, the answer for the example question given earlier should not only contain the answers such as "whale," "sperm whale," and so on but also should explicitly indicate that "whale" is "the largest animal on earth." Instead of directly annotating the recognized transcripts, however, we used the corresponding video frames for evaluation because users can directly find the answers in both the retrieved video clips and recognized text and because the OCR document is not as perfect as human-typed text. Hence, a passage was labeled as positive if and only if its closed caption contained the answer. In summary, 140 of the 500 questions did not have an answer. Meanwhile, 639 answer candidates were labeled as true answers. On average, there are 1.27 labeled answers for each question.

Collection Comparison

We compare this work to five related studies that had presented evaluations for the overall videoQA performance. The comparison shown here involves three parts: raw video collection size, question set, and answer assessment. Some studies were excluded here since they did not explicitly present the detailed information of the used video data and question set used. The TREC-VID task also was disregarded here since the target is quite different. The TREC-VID task focuses on retrieving shots or clips with structural queries (e.g., “find shots of a person walking or riding a bicycle,” where the underlying words could be replaced by the other terms). Table 3 presents the comparison of related works.

As shown in Table 3, both the question and document collections used in this work are substantially larger than those of previous works (Cao & Nunamaker, 2004; Cao et al., 2005; C.J. Lin et al., 2001; Wu et al., 2004; Yang et al., 2003; Zhang & Nunamaker, 2004). In terms of question sources, our question set is much larger and closer to actual real-world user queries. As noted by Voorhees (2000, 2001), questions constructed explicitly for special purposes are often unnatural and also make the QA task easier. This is because questions derived from reading the content of a document usually can be easily retrieved. In other words, we suggest that a sizable experimental question set yields more reliable experimental results.
TABLE 3. A video-collection comparison with previous literature.

<table>
<thead>
<tr>
<th>Document collection</th>
<th>Question collection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Previous literature</td>
<td>Size of data</td>
</tr>
<tr>
<td>Lin et al. [21]</td>
<td>6 hr</td>
</tr>
<tr>
<td>Yang et al. [44]</td>
<td>6 hr</td>
</tr>
<tr>
<td>Cao et al. [5, 6]</td>
<td>N/A</td>
</tr>
<tr>
<td>Wu et al. [42]</td>
<td>6 hr</td>
</tr>
<tr>
<td>Zhang and Nunamaker [47]</td>
<td>20 hr</td>
</tr>
<tr>
<td>This article</td>
<td>130 hr</td>
</tr>
</tbody>
</table>

FIG. 2. Overview of the multilingual video question answering systems.

results. In terms of answer assessment, most literature does not explicitly describe how answers were judged.

Video Text Processing for VideoQA

An overview of our videoQA system can be found in Figure 2. The video-processing components transform input videos into OCR documents as the first stage. We then transform the OCR documents into English with a machine translation tool. In fact, it is possible to instead directly translate input questions into Chinese rather than translate the answer document into English. But in preliminary experiments, it was found that document translation is not only more effective than query translation but also simultaneously yields both English and Chinese outputs.

When the input question does not contain Chinese characters, our system will view the question as an English question and perform English videoQA. In English, we apply common word-tokenization techniques and remove punctuation, and stopwords. In Chinese, we segment the Chinese into unigram, bigram, and trigram words. For example the term “亚历山大” (Alexander) will be tokenized as: (unigram) “亚(yia),” “历(li),” “山(shan),” “大(da);” (bigram) “亚历(yia+li),” “历山(li+shan),” “大(shan+da);” (trigram) “亚历山(yia+li+shan)” and “历山大(li+shan+da).” One can employ a well-known Chinese word segmentation tool to replace the fixed-length grains.

To increase the efficiency of our method, we adopted the Okapi BM-25 retrieval method (Over et al., 2007; Robertson et al., 2000) to retrieve the top-1,000 passages as the input data for our ranking modules. Finally, the proposed ranking algorithm retrieved top-N passages, along with their corresponding video clips, as answers in response to the question.

Video Processing

Here, the video-processing module takes a video and recognizes the closed-caption texts. An example of the input and output associated with the whole video-processing component can be seen in Figure 3. The video-processing module consists of four important steps: text localization, pixel binarization, frame tracking, and OCR. First, a set of frames (sampled at two frames/s) were extracted from videos. We then employed the edge-based filtering method (Lyu, Song, & Cai, 2005; Sasaki, Lin, Chen, & Chen, 2007) to eliminate the non-text pixels. Next, the modified coarse-to-fine, top-down block segmentation algorithm (Lienhart & Wernicke, 2002) was used to precisely locate the text components. Then, following Lyu et al.’s (2005) approach, we slightly modify the binarization method to remove the non-text pixels in each text component. Before performing OCR, to remove

---

1http://www.systran.co.uk/
We also adopted the word reranking methods (C.J. Lin et al., 2001) (Strategy 3) to improve the OCR errors. All of the aforementioned steps can be realized without annotated training data. The video-processing component can be viewed as a black box that automatically recognizes text from the input videos. Here, we adopted the video OCR methods from previous literature (Lee et al., 2005; C.J. Lin et al., 2001; Wu et al., 2004). We also adopted the word reranking methods (C.J. Lin et al., 2001) (Strategy 3) to improve the OCR errors.

We sampled 30 video clips (1,684 frames) to validate the text detection and extraction performance and randomly selected 10 clips to evaluate the overall OCR accuracy. In our experiments, the text-localization module produces recall and precision rates of 98.93 and 97.63%, respectively. In terms of the overall OCR accuracy, our videoOCR module achieves an \( F \)-measure of 84.74. The overall processing time for a 52-min video segment was less than 18 min.\(^2\)

In this article, we treat all words that appear in the same frame as one sentence. 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. For each OCR document, we grouped every three continuous sentences with one previous sentence, overlapped to form a passage.

Even if a passage constructed from sentences at the boundaries of two semantic fragments might be inconsistent, our main concern was whether the three sentences would contain the answers. The answer string is usually very short. For fact-based (i.e., factoid) questions (e.g., “Where was Napoleon born?”), the passage will be viewed as positively related if the passage contains sufficient information and is a true answer in response to the question. For example, the passage “When Napoleon arrived in Waterloo, he was 46 years old. Napoleon was born in Corsica on August 8, 1769 . . . .” is judged as a positive result.\(^3\)

:\(^2\)Hardware information: Intel 2.8 GHz with 2 GB RAM under the WindowsXP OS

:\(^3\)More complete example can visit our online demonstration Web site [in Chinese]: http://140.115.112.118/bcbb/TVQS3/Question/index.htm

The Proposed Passage-Retrieval Algorithms

The ranking model receives the segmented passages from previous steps and outputs the top-ranked passages to respond to the question. Tellex et al. (2003) compared seven passage-retrieval algorithms such as Okapi BM-25 (Robertson et al., 2000; Savoy, 2005) and SiteQ (G.G. Lee et al., 2001) for the TREC-QA task, excluding two ad hoc methods which required human-generated patterns and ontology. They reported that the term-weighting methods such as BM-25 were slightly worse than was the SiteQ’s approach when integrating with external resources, including a named entity recognizer, WordNet, and a thesaurus. More recently, both Cui, Sun, Li, Kan, and Chua (2005) and Shen and Klakow (2006) stated that by using syntactic parsers, the passage-retrieval algorithms achieved state-of-the-art performance. For example, Cui et al. showed that their fuzzy relation syntactic matching method substantially outperformed the SiteQ method with up to \( \sim 78\% \) relative greater performance. But the limitation of those approaches is that it requires the dependency parsing information and sizable training data as compared to the retrieval-based approaches. In many Asian languages such as Chinese and Japanese, parsing is much more difficult since it needs to solve the word-segmentation problem before part-of-speech tagging, chunking, and parsing (Charniak, 2000; Collins, 1998). Furthermore, the development of a thesaurus or labeled training dataset for QA is laborious and time consuming. Compared to Cui et al.’s method, traditional term-weighting models are much less costly, more portable, and more practical.

OCR documents are not as good as traditional text articles which have been correctly typed manually. A portion of the words found were error-recognized, unrecognizable, or a false-alarm. These unexpected words significantly affect the Chinese word segmentation process as well as parsing since words are incorrectly tokenized in earlier stages. In our experiments (see Table 4 and Table 5), we also showed that the use of a well-trained, high-performance, Chinese word segmentation tool gave considerably worse results than did using atomic Chinese characters for all passage-ranking models.

However, traditional term-weighting methods often give more weight to the passages that potentially contain some high-frequency words. Even though the SiteQ’s models take word distribution into account, they still prefer high-frequency terms if the passage tends to include abundant keywords rather than \( N \)-gram chunks (see Equations 1 and 2).
in G.G. Lee et al., 2001). Usually, the \( N \)-gram information is much more important than are high-frequency unigrams. For example, a passage that contains the three words “multimedia,” “information,” and “retrieval” often receives a similar score to a passage which has the trigram “multimedia ∩ information ∩ retrieval.” It is often the case that an \( N \)-gram chunk is much more unambiguous than its individual unigrams. Thus, we attempt to put more emphasis on \( N \)-gram matches and also take the word density into account.

To effectively make use of both \( N \)-gram and keyword match-density information, we propose a ranking algorithm that integrates the information from both of them. Roughly speaking, our method seeks to find the passage that contains both high-density and long \( N \)-gram match words. The high-density term distribution supposes that the dense keywords might contain answers while long \( N \)-gram match words preserve more lexical information from the question. We furthermore restrict the term matching to be a one-to-one mapping (i.e., mapping each question word to a specific position of the passage) instead of a one-to-many mapping that considers all match positions. In this way, our method is prevented from overemphasizing high-frequency words. In other words, it searches for a match sequence that produces the most dense and long \( N \)-gram match. The match sequence is then used to compute the ranking score. In this section, we first review our previously proposed “finding the optimal string pattern” (FOSP) algorithm to extract dense and long \( N \)-gram match string patterns. Based on the observed matched strings, we present the designed scoring functions for computing the passage score.

### FOSP Algorithm

Before introducing the FOSP algorithm, we give the following notations. Assume the Passage \( P \) contains \( T \) words denoted as \( P = PW_1, PW_2, \ldots, PW_T \), and \( Q = QW_1, QW_2, \ldots, QW_T \) denotes the sequence of words in Question \( Q \). A common subsequence for Passage \( P \) can be expressed as \( \text{Sub}_P = PW_1, PW_{k+1}, \ldots, PW_{k+y-1} \) and if \(|\text{Sub}_P| = x \). Similarly, \( \text{Sub}_Q = QW_k, QW_{k+1}, \ldots, QW_{k+y-1} \) indicates a subsequence match for \( Q \) if \(|\text{Sub}_Q| = y \). A common subsequence represents a continuous string matching between \( P \) and \( Q \). \( \text{Sub}_P \) means the \( i \)th matched continuous string (i.e., common subsequence) in the passage while \( \text{Sub}_Q \) indicates the \( j \)th matched continuous string in the question. Figure 4 gives an example of common subsequences for \( P \) and \( Q \).

As shown in Figure 4, we exclude the OCR-unrecognizable words (i.e., \(<URW>\)) from \( P \) and stopwords from \( Q \) since they do not help answer identification. The stopword identification is based on a greedy maximum matching algorithm. To collect the stopwords in Chinese, we first tokenize the Chinese words into unigram, bigram, and trigram levels and count the term frequency in the collected million-Chinese knowledge base\(^4\) and news articles.\(^5\) Terms with high-frequency (>100) or low-frequency (<5) words were preserved. A native expert then reviewed and manually added these terms into the list. Finally, we integrate the 571 well-defined English stopwords\(^6\) found in the list. In total, 897 English and Chinese words were stored in our common word list.

The FOSP algorithm aims at finding the optimal match sequence in \( P \) which produces high-density and long \( N \)-gram match. Unfortunately, finding such an optimum solution is an NP-hard problem (Cormen, Leiserson, Rivest, & Stein, 2002). To solve this, an approximate solution is designed to find the optimal matched strings (for more detail of the algorithm, see Wu, Lee, Yang, & Yen, 2006).

---

\(^4\)http://tw.knowledge.yahoo.com/
\(^5\)http://udn.com/NEWS/main.html
\(^6\)http://www.lextek.com/manuals/onix/stopwords2.html
FIG. 4. A preprocessing algorithm for finding the possible match positions in the passage.

Given a passage:

<table>
<thead>
<tr>
<th>PW1</th>
<th>PW2</th>
<th>PW3</th>
<th>PW4</th>
<th>PW5</th>
<th>PW6</th>
<th>PW7</th>
<th>PW8</th>
<th>PW9</th>
<th>PW10</th>
<th>PW11</th>
<th>PW12</th>
<th>PW13</th>
<th>PW14</th>
<th>PW15</th>
</tr>
</thead>
<tbody>
<tr>
<td>PW1</td>
<td>PW2</td>
<td>PW3</td>
<td>PW4</td>
<td>PW5</td>
<td>PW6</td>
<td>PW7</td>
<td>PW8</td>
<td>PW9</td>
<td>PW10</td>
<td>PW11</td>
<td>PW12</td>
<td>PW13</td>
<td>PW14</td>
<td>PW15</td>
</tr>
</tbody>
</table>

By removing the stop-words, the passage string can be converted as:

\[ P = PW_1, PW_2, PW_3, \ldots, PW_{10} \]

Given a question:

| What will be used to replace the existing space shuttles? |
| stopword | stopword | stopword | stopword | stopword | stopword |

By removing all stopwords, the question string should be:

\[ Q = QW_1, QW_2, QW_3, QW_4 \]

By exhaustive matching words between \( Q \) and \( P \), all the common subsequences for \( P \) are:

\[ \text{Sub}_P^1 = PW_1, PW_6, PW_9 \quad \text{Sub}_P^2 = PW_{18}, PW_{19}, PW_{20}, PW_{21}, PW_{22}, PW_{23} \]

Similarly the common subsequences for \( Q \) are:

\[ \text{Sub}_Q^1 = QW_1 \quad \text{Sub}_Q^2 = QW_2, QW_3 \quad \text{Sub}_Q^3 = QW_6, QW_7, QW_8 \]

String Pattern Weighting

After extracting the optimal subsequences, we compute the following score to rank passages.

\[
\text{Passage\_Score}(P) = (1 - \lambda) \cdot \text{W\_density}(Q, P) + \lambda \cdot \text{W\_weight}(Q, P) \\
= (1 - \lambda) \cdot \frac{\text{length}(\text{Sub}_Q^1)}{\text{length}(P)} \times \frac{\text{TF}(\text{Sub}_Q^1, P)}{\text{K} + \text{TF}(\text{Sub}_Q^1, P)}
\]

The first term of Equation 2 estimates the question word density degree in Passage \( P \) while “\( \text{W\_weight}(Q, P) \)” measures the matched question word weights in \( P \). \( \lambda \) is a parameter, which is used to adjust the importance of the \( \text{W\_weight}(Q, P) \). Both of the two estimations employ the subsequence information for \( P \) and \( Q \). We will separately introduce the computation of \( \text{W\_density}(Q, P) \) and \( \text{W\_weight}(Q, P) \).

The \( \text{W\_density}(Q, P) \) is designed to quantify “how dense the matched question words in Passage \( P \)”. It also takes the term weight into account. By extracting common subsequences in the question, the set of \( \text{Sub}_Q^1 \) can be used to measure the question word density. At the beginning, we define Equation 2 for weighting \( \text{Sub}_Q^1 \).

\[
\text{Weight}(\text{Sub}_Q^1) = \text{length}(\text{Sub}_Q^1)^{\alpha_1} \times \text{DP}(\text{Sub}_Q^1)
\]

where \( \text{length}(\text{Sub}_Q^1) \) is merely the length of \( \text{Sub}_Q^1 \) (i.e., number of words in \( \text{Sub}_Q^1 \)). \( \alpha_1 \) is a parameter that controls the weight length for \( \text{Sub}_Q^1 \). The second term in Equation 2 estimates the “discriminative power” (DP) of the subsequence. Some important, but low-frequency, words should be given more weight than should some common words. To measure the DP score, we extend the BM-25 term-weighting schema.

\[
\text{DP}(\text{Sub}_Q^1) = W' \times \frac{(k_1 + 1) \times \text{TF}(\text{Sub}_Q^1, P)}{K + \text{TF}(\text{Sub}_Q^1, P)}
\]

\[
+ \frac{(k_3 + 1) \times \text{TF}(\text{Sub}_Q^1, Q)}{k_3 + \text{TF}(\text{Sub}_Q^1, Q)}
\]

\[
W' = \log\left( \frac{N_p - \text{PF}(\text{Sub}_Q^1) + 0.5}{\text{PF}(\text{Sub}_Q^1) + 0.5} \right)
\]

\[
K = (1 - b) + b \times \frac{|P|}{\text{AVG}(|P|)}
\]

\[
k_1, k_3, \text{ and } b \text{ are constants, which are set as 1.2, 0.75, and 500, respectively (Over et al., 2007; Robertson et al., 2000).}
\]

\[
\text{TF}(\text{Sub}_Q^1, Q) \text{ and } \text{TF}(\text{Sub}_Q^1, P) \text{ represent the term frequency of } \text{Sub}_Q^1.
\]

Equation 5 computes the inverse “passage frequency” (PF) of \( \text{Sub}_Q^1 \) as opposite to the traditional inverse “document frequency” (DF), where \( N_p \) is the total number of passages. The collected video database is a small, but “long,” OCR document set, which will cause the estimated DF value to be unreliable; however, a passage is actually...
Algorithm 1: Retokenizing_a_subsequence

Input: A subsequence $Sub_Q^j$ where $start_j$ is the position of first word of $Sub_Q^j$ in question and $end_j$ is the position of last word of $Sub_Q^j$ in question

Output: A set of retokenized subsequence $\{TSub_1, TSub_2, \ldots, TSub_{N_t}\}$ $N_t$: the number of retokenized subsequence

Algorithm:

Initially, we set $N_t := 1$; $TSub_1 := QW_{start_j}$

if ($Sub_Q^j \neq \emptyset$)

/* from the start to the end positions in the string */

$Prev_word := QW_{start_j}$

for ($k := start_j + 1$ to $end_j$) {

/* Check the two question words is bigram in the passage */

if (bigram ($Prev_word$, $QW_k$) is_found_in_passage)

add $QW_k$ into $TSub_{N_t}$

else {

$N_t$++;

$TSub_{N_t} := QW_k$

} /* End else */

$Prev_word := QW_k$

} /* End for */

else{

$N_t := 0$;
}

FIG. 5. An algorithm for retokenizing subsequence.

Algorithm 2: Computing_DP_score

Input: A subsequence $Sub_Q^j$ where $start_j$ is the position of first word of $Sub_Q^j$ in question; $end_j$ is the position of last word of $Sub_Q^j$ in question

Output: The score of $DP(Sub_Q^j)$

Algorithm:

if (($end_j - start_j) \geq 2$)

Head := $end_j - 2$;

else if (($end_j - start_j) \geq 1$)

Head := $end_j - 1$;

else

Head := $end_j$;

tail := $end_j$;

Max_score := 0;

for ($k := head$ to $tail$) {

let $WORD := QW_k, QW_{k+1}, \ldots, QW_{tail}$;

/*** look-up $WORD$ in the index files ***/

compute $DP(WORD)$ using equation (4);

if ($DP(WORD) > Max_score$)

$Max_score := DP(WORD)$;

} /* End for */

$DP(WORD) := Max_score$;

FIG. 6. An algorithm for computing discriminative power (DP) score for a subsequence.

more coherent than is a long document; thus, we replace the DF estimation with the PF score. Note that $Sub_Q^j$ might be a long string that needs to be further retokenized into finer grained subsequences which are closer to the actual words. We therefore propose two algorithms to: (a) retokenize an input subsequence and (b) compute the DP score for a subsequence. Note that it is impossible to compute the DP score for long strings since we only kept the uni-/bi-/trigrams information. Figure 5 and Figure 6 list the proposed two algorithms.
The proposed Algorithms 1 and 2 can be used to tokenize and compute the DP score of not only \( \text{Sub}_j^Q \) for question but also \( \text{Sub}_j^Q \) for passage. As seen in Figure 5, it requires DP information for different lengths of \( N \)-gram. As noted earlier, the unigram, bigram, and trigram levels of words were stored in indexed files for efficient retrieving and computing of the DP score at this step. One can further consider four-grams, five-grams, and so on, but we found that the use of more than trigrams does not significantly improve the performance in our experiments.

By applying Algorithm 1 to the set \( \text{Sub}_j^Q \), we can obtain all retokenized subsequences (\( \text{TSub}_j^i \)). We then use the retokenized subsequences to compute the final density score. Equation 6 specifies the designed Density scoring function.

\[
W_{\text{Density}}(Q, P) = \sum_{i=1}^{T_{\text{CNT}}-1} \frac{\text{Weight}(\text{TSub}_j^i) + \text{Weight}(\text{TSub}_j^{i+1})}{{\text{dist}(\text{TSub}_j^i, \text{TSub}_j^{i+1})}^{\alpha_2}} \tag{6}
\]

\[
\text{dist}(\text{TSub}_j^i, \text{TSub}_j^{i+1}) = \min_{\text{P}} \text{distance between}(\text{TSub}_j^i, \text{TSub}_j^{i+1}) + 1 \tag{7}
\]

\( T_{\text{CNT}} \) is the total number of retokenized subsequences, which can be extracted through applying Algorithm 3 for all \( \text{Sub}_j^Q \). Equation 7 merely counts the minimum number of words between two neighboring retokenized subsequences \( \text{TSub}_j^i \) and \( \text{TSub}_j^{i+1} \) in the passage. \( \alpha_2 \) is the parameter that controls the impact of distance measurement. A passage with a highly dense \( \text{TSub}_j \) distribution usually receives a high-density score.

The density scoring can be regarded as measuring “how much information the passage has corresponding to the question.” On the contrary, the Weight (second term in Equation 1) aims to estimate “how much content/information the passage has given the question.” To achieve this, we take the other extracted common subsequences (i.e., \( \text{Sub}_j^P \)) into account. By means of the same term-weighting schema for the set of \( \text{Sub}_j^P \), the Weight is then produced. Equation 8 gives the overall Weight measurement.

\[
W_{\text{Weight}}(Q, P) = \sum_{i=1}^{S_{\text{CNT}}} \left[ \text{length}(\text{Sub}_j^P)^{\alpha_1} \times \text{DP}(\text{Sub}_j^P) \right] \tag{8}
\]

where the DP score of the input subsequence can be obtained via Algorithm 2 (Figure 6). \( S_{\text{CNT}} \) is the number of subsequences in P. The parameter \( \alpha_1 \) also is set equal to its setting in Equation 2.

Thus far, the \( W_{\text{Density}} \) and \( W_{\text{Weight}} \) have been introduced. By combining the two measurements with different weights (\( \lambda \)), the final passage score is therefore determined. The top-ranked answer is the passage that receives the highest score according to Equation 1.

In our previous method (Wu & Yang, 2007), the passage score is computed by stressing either head or tail sentences in the passage; however, it is usually the case that the answer is a substring of the passage. Such a head–tail weighting scheme cannot tolerate the errors produced by the OCR modules. For example, some blurred images will generate incorrect text information. Therefore, rather than compute the whole passage score, we select the optimal scores from each subset of the passage. In other words, we compute Equation 1 for each subset in the passage. That is,

\[
P = \{S_1, S_2, \ldots, S_K\}
\]

\[
\text{Passage Score}(P) = \max_i \{\lambda \times W_{\text{Density}}(Q, C_i) + (1 - \lambda) \times W_{\text{Weight}}(Q, C_i)\} \tag{9}
\]

where \( C_i \in 2^K - 1 \)

\( K \) denotes the number of sentences in P. Here, we define the passage as consisting of a set of sentences. Then, the subset is chosen by considering all combinations of sentences in the passage. For example, if \( |P| = 2 \), then the passage score is obtained by computing Equation 1 over \( S_1, S_1 + S_2, \) and \( S_2 \).

Obviously, our method is not as efficient as are traditional ranking models such as BM-25, particularly when the passage size scales upward. As noted previously, to reduce the unnecessary computation required for most irrelevant passages, we further employed the top-1,000 passages retrieved from BM-25 as input to our method. This technique greatly reduces the computational costs for the low-ranked passages. Hence, the proposed passage-ranking algorithm also can be viewed as a reranking for the initial retrieval models. By utilizing the initially retrieved top-1,000 passages from BM-25, the whole response time of the implemented videoQA system is under 1 s in responding to a question.7

Experiments

In this section, we present quantitative evaluations using real-world videos with our method. To compare with the other ranking algorithms, we employed seven top-performing approaches which are language and domain portable for comparison. Next, we briefly introduce the evaluation metrics, then the overall experimental results are presented. Finally, we describe parameter validation for the proposed method.

Evaluation Metrics

The main reciprocal rank (MRR) score is normally used (see Cao & Nunamaker, 2004; Cao et al., 2005; Dang et al., 2007; Ittycheriah, Franz, & Roukos, 2001; Voorhees, 1999, 2000, 2001; Yang et al., 2003) to evaluate overall QA performance, which is defined as Equation 10.

\[
MRR = \frac{1}{N_Q} \sum_{i=1}^{N_Q} \frac{1}{\text{rank}_i} \tag{10}
\]

where \( N_Q \) is the number of testing questions, and \( \text{rank}_i \) is the first correct answer for Question \( i \). If the answer is found

---

7Our platform was based on WindowsXP, single Opteron 2.2 GHz processor with 2.0 GB RAM
at multiple ranks, the best rank will be used. If no relevant answer is found in the rank list, the score of this question is set to zero. In addition, the precision rate measures the ratio of correct answers found in the whole ranked list while the pattern recall rate estimates the ratio of correct predicted answers divided by the number of labeled answer patterns. Equations 11 and 12 define the formulae for computing the precision and pattern recall rates.

\[
\text{Precision} = \frac{\text{No. of correct predicted answers}}{\text{No. of predicted answers}} \quad (11)
\]

\[
\text{Pattern Recall} = \frac{\text{No. of correct predicted answers}}{\text{No. of actual labeled answer patterns}} \quad (12)
\]

Both the numerators in Equations 12 and 13 are identical and represent the number of correctly predicted answers in the retrieved passages. In this article, we measure the MRR scores for both the top-one and the top-five ranks, and the precision and pattern-recall rates for the top-five retrieved answers. By means of the aforementioned three measurements, a relative percentage rate can be calculated to measure the improvement resulting from using a better method. The relative percentage score is defined as follows.

\[
\text{Relative percentage} = \frac{|\text{measure}_1 - \text{measure}_2|}{\text{measure}_2} \times 100\% \quad (13)
\]

where measure$_1$ and measure$_2$ are one of the measurements of either MRR, precision, or pattern-recall rates for Methods 1 and 2. For example, if the MRR score of the language model is 0.515, and 0.540 for our method, we say that our method relatively outperforms the language model by \((0.540 - 0.515)/0.515 \times 100\% = 4.85\%\).

**Experimental Results**

In this article, we employed seven top-performing, yet portable ranking, models: TFIDF, Okapi BM-25 (Robertson et al., 2000; Savoy, 2005), INQUERY (Broglio et al., 1995; Ponte & Croft, 1998), the language model (Ponte & Croft, 1998; Zhai & Lafferty, 2001, 2002), cosine, and SiteQ's$^8$ (G.G. Lee et al., 2001) approaches, and Akida’s model (Akiba, Fujii, & Itou, 2004) for comparison. For a fair comparison, we also used the same preprocessing steps (i.e., stopword removal and tokenization) for these methods. For the language model, a two-stage smoothing method was employed (with the Dirichlet prior \(\mu = 200\) and interpolation parameter \(\lambda' = 0.1\), which were selected through validation experiments (see Appendix).

The overall videoQA performance was evaluated with 500 natural language questions. The system performance was evaluated through the top-five passages returned. For the parameter settings, we set \(\alpha_1 = 1.2\), \(\alpha_2 = 0.5\), and \(\lambda = 0.6\), which were found via parameter validation (discussed earlier). Table 4 lists the whole set of videoQA results with different ranking models.

The experimental results clearly demonstrate that of all the retrieval algorithms, this article’s method achieves the best system performance and is slightly better than the language model in terms of the nonanswered question rate (177 vs. 187). It produced scores of 0.518 and 0.569 for MRR when evaluating the top-one and top-five passages, respectively, while the precision rate is 0.192. Compared with the second-best method (language model), this article’s method is 11.20% better (relative percentage) in terms of the MRR (top-one) score. For the MRR (top-five) score, our method is 8.36% better (relative percentage). In comparison to the third-best ranking model (BM-25), this article’s method is 20 and 14.25% better in the MRR (top-one) and MRR (top-five) scores, respectively, and 11.26 and 10.93% better in terms of precision and pattern-recall rates, respectively. In terms of the answer coverage rate, this article’s method is able to answer more questions (500 – 177 = 323) compared to the other ranking models. Overall, the experiment shows that this article’s method outperforms the other seven methods in terms of all the evaluation metrics.

Next, we replace the unigram-level approach to Chinese word that was tokenized by the trained Chinese word tokenizer. Parsers and part of speech taggers were developed based on “words.” This experiment was conducted to evaluate the impact of shallow natural language processing (NLP) components. In this experiment, we employed a Chinese word segmentation tool (Wu, Yang, & Lin, 2006) that achieved a \(\sim\)0.93 F-measure in the SIGHAN bake-off tasks (Levow, 2006). Table 5 lists the overall experimental results using the word-level approach. In comparison to the unigram-level approach (see Table 4), it is shown that the use of the word segmentation tool does not improve and, in fact, decreases the videoQA result for most ranking models. For example, the relative performance of the language model is 11.53 and 10.92% worse in MRR (top-one) and MRR (top-five) scores, respectively. In terms of precision and pattern-recall rates, it drops 13.66%, and in terms of relative percentages, it drops 13.93%. For our method, the MRR score substantially decreases by more than 10% for all rates measurements except the “precision value.”

The segmented words did not allow partial matches at the unigram level. It strictly forces an exact word match. Therefore, it is more difficult to match a question word with a passage word since mismatching inevitably occurs with OCR error-recognized or false-alarm words. For example, “太空梭” (space shuttle) cannot match the word “大(big) 空梭 (shuttle)” in the OCR document. Consequently, we can see that the use of high-performance word-segmentation tools under noisy data (OCR document) considerably and negatively affects the ranking performance of these approaches.

Third, we evaluate the English/Chinese bilingual videoQA results. As defined in this article, we test English to Chinese (E–C) and English to English (E–E) tasks. Task E–E is
used to evaluate the performance of English query to English documents (OCR transcripts). This can be viewed as the document translation for English input questions. By following most bilingual QA and information-retrieval tasks, such as NTCIR-CLQA (Sato, Kanade, Hughes, Smith, & Satoh, 1999) and CLEF@QA (Sasaki et al., 2007), the questions are parallel in English and Chinese. To achieve this, we manually translate the Chinese questions into English. On the other hand, we employ the online machine translation tool10 converting Chinese OCR transcripts into English. Table 6 lists the overall results of the E–E task. In this task, our method clearly outperforms the other approaches.

For the E–C task, we adopt the same translation tool to transform the English question into Chinese to retrieve the Chinese OCR transcript. The task also can be treated as query translation for English input questions. Users could directly access the Chinese videos with English through converting the query into Chinese. Here, we also evaluate the effects of the character level and the word level of words. Tables 7 and 8 list the results of the E–C tasks by using both the character level and the word level.

By contrast to the E–E task, the performance of query translation (E–C) is much better than the document-translation strategy (E–E). Meanwhile, the character level again outperforms the word-level chunk. On the basis of the E–C and E–E tasks, the optimal solution for supporting a bilingual videoQA system is to directly translate the English query into Chinese to retrieve the Chinese videos. We also perform the statistical test to evaluate the significant test among all the retrieval methods. Table 9 summarizes the significant analyses (t test) between our method and the other retrieval methods.

10Systran 6.0 http://www.systran.co.uk/translation-products/desktop/systran-premium-translator

Table 9. Statistical significant analysis between our method and the other retrieval methods.

<table>
<thead>
<tr>
<th></th>
<th>C–C</th>
<th>E–C</th>
<th>E–E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language Model</td>
<td>0.076***</td>
<td>0.094***</td>
<td>0.242*</td>
</tr>
<tr>
<td>BM-25</td>
<td>0.007***</td>
<td>0.017***</td>
<td>0.004***</td>
</tr>
<tr>
<td>INQUERY</td>
<td>0.003***</td>
<td>0.005***</td>
<td>0.226*</td>
</tr>
<tr>
<td>TFIDF</td>
<td>0.006***</td>
<td>0.015***</td>
<td>0.098**</td>
</tr>
<tr>
<td>Akida’s</td>
<td>0.000***</td>
<td>0.003***</td>
<td>0.000***</td>
</tr>
<tr>
<td>SiteQ</td>
<td>0.000***</td>
<td>0.000***</td>
<td>0.000***</td>
</tr>
<tr>
<td>Cosine</td>
<td>0.000***</td>
<td>0.000***</td>
<td>0.000***</td>
</tr>
</tbody>
</table>

*p > .1. **p < .1. ***p < .05.
It produces slightly a significant difference in comparison to the language models.

**Discussion**

The proposed method is based on weighting the optimal matched sequence for the question and passage. This method is very effective when the key terms of the question belong to the phrase type (e.g., when the question “What is the largest gambling house in Las Vegas?” asks for the name of the “largest gambling house”). According to Equation 3, once the passage contains this phrase, the obtained score is $3 \times 3 + 3 \times 3 = 18$. On the contrary, if another passage only matched a partial fragment of a phrase (i.e., “largest animal”), it is weighted with a far smaller score. This method is especially suitable for OCR documents. Although some key characters or words could not be recognized by our OCR method, the correct answers were retrieved successfully. For example, the first question in Figure 7 asks for the landmark skyscraper in New York. The top-ranked answer completely matches six Chinese characters and achieves the best score. The top-ranked answer is “曼哈顿的地标” [Chinese: New York City]. Notice that the passage contains two OCR errors in the Chinese characters (the underlined characters). If the matched subsequence is long enough to capture the key question terms, it still could be weighted higher, similarly for the other two questions in Figure 7. For the three questions, our videoQA system finds the correct answers at the first rank.

In essence, our method places more weight upon the passages that contain “long” and “high dense” N-gram match; however, it frequently fails in the following cases:

- OCR errors in key question words
- Synonyms and anaphora
- Lack of language-dependent syntax and semantic analysis
- Machine Translation errors

Although our method can handle OCR documents, when important question words cannot be correctly identified, the passage score is discounted. Take, for example, the question “Where is FBI?” Unfortunately, most “FBI” terms were recognized as “28I” or “98I” by our video OCR. These incorrectly recognized terms were significant in causing this study’s method to fail to find the key important word, “FBI.”

Another limitation is the synonym and anaphora problems. This article’s method focuses on matching the surface terms, but when the keywords were replaced by pronouns or synonym terms, the actual answers could not be found.

The third problem of this approach is that it does not take the language-dependent characteristics into account. In Asian languages such as Chinese, there is often not a specific symbol between words. If the sentence was not tokenized, our method would match the wrong word subsequence. For example, the Chinese question “最早的掠食者出现在何时?” [Chinese: When did the earliest predator appear on earth?] attempts to find the earliest predator. Without losing generality, we use atomic
Chinese characters to represent the previous question as follows.

A B C D E F G H I J

Each A, B, C . . . J represents its corresponding Chinese character. In real-world use, this Chinese sentence should be tokenized as

AB C DEF GH IJ

where C is a Chinese stopword that should be excluded. However, the returned answers extract false subsequences, which matched with CDEF. This will highlight the incorrect answers in the higher scored ranks. In comparison, passages which contained actual keywords AB and DEF obtained lower scores. In this case, a language-dependent component is useful to overcome this problem.

The final limitation is only for bilingual QA purposes. In this article, we employed a commercial machine translation tool to automatically convert the Chinese OCR documents into English. It is known that some proper nouns could not successfully be translated into English due to the translation software lacking sufficient vocabulary. If these words cannot be correctly translated, the actual answers will not be found. For instance, the question “Who is Hatshepsut’s father?” asks for the name of Hatshepsut’s father; however, the name Hatshepsut was translated from Chinese characters into the phrase “conspicuous Zhai” in English. Therefore, it resulted in the real answers being missed.

Parameter Validation

We randomly select 150 questions for parameter validation. There are three parameters: λ, α₁, and α₂ within our ranking algorithm. λ adjusts the weight of the W_Weight (Q, P) in Equation 2, and should be in a range from 0 to 1. α₁ and α₂ control the term length and distance impacts in Equations 3 and 7. In view of our retrieval model, it is necessary to first identify α₁ or α₂ and then λ however the search space for α₁, and α₂ is quite large and usually intractable. We therefore adopted the “local optimum search” (LOS) strategy to find the best parameter settings with fixed local steps. For example, for the line of λ = 0.2 in Figure 8, we plot each point by searching for the optimal α₂ value for its corresponding α₁. If no better α₂ is found in the next three steps, then the best α₂ is selected for this α₁ without future searching.

As shown in Figure 8, it is clearly the case that a slight weight on α₁ usually performs better than not using weighting (i.e., set α₁ = 1). Even simply setting α₁ = 1, the technique can still achieve very satisfactory results; however, over-weighting (set α₁ > 1.4) or underweighting (α₁ < 1) this factor causes a relative performance decrease of almost 10% in the MRR score. In other words, taking the length of the N-gram into consideration is very useful for ranking.

In the second experiment, we further restricted the search space for α₁ to the range 1~1.7. Similarly with previous results, it was found that when λ = 0.5, a better system performance is produced. In this experiment, the LOS usually stopped at α₁ = 1.4 for each α₂. As seen in Equations 7 and 8, α₂ gives a discount to the number of words between two subsequences. This means that we should reduce the ease of distance measure in Equation 8, where the simple setting for α₂ is not a good choice since the larger the α₂, the worse results it attracts. As shown in Figure 9, we can see that the optimal settings for α₂ should be restricted between 0.4 and 0.6 (≈0.5). This implies that we could compute the square root for the distance measure for Equations 7 and 8; that is,
we can rewrite Equation 7 as:

$$W_{\text{Density}}(Q, P) = \sum_{i=1}^{T_{\text{CNT}}-1} \frac{\text{Weight}(T_{\text{Sub}_i}) + \text{Weight}(T_{\text{Sub}_{i+1}})}{\sqrt{\text{dist}(T_{\text{Sub}_i}, T_{\text{Sub}_{i+1}})}}$$

Next, we use the verified $\alpha_1(=1.4)$ and $\alpha_2(=0.4\sim0.6)$ settings to peak $\lambda$ as shown in Figure 10. In this experiment, we evaluated the results for these groups using the whole testing set.

We continue our experiment to validate the impact of different passage size. Figure 11 plots the performance of variant passage size. The row in Figure 11 represents the sentence count in a passage. The value is the optimally selected among all its possibilities. For example, for the point of sentence count is 3, we select the optimal MRR score from $0\sim3$, $1\sim3$, or $2\sim3$, where $1\sim3$ means that the passage contains two new sentences and one previous sentence. Clearly, our method achieves better MRR scores than do the other retrieval approaches. The best MRR score (0.637) is reached by segmenting the passage with one previous and four new sentences (i.e., 1–5).

On the other hand, we analyze the impact of the number of initial retrieved passages from different ranking models. In this experiment, we use the whole question set to see the improvement over the four top-performing retrieval models (including language models, BM-25, TFIDF, and INQUERY). We use the top-$N$ retrieved passages from the
TABLE 10. MRR improvement rate (%) over the selected four initial retrievers.

<table>
<thead>
<tr>
<th>Initial retrievers</th>
<th>TFIDF</th>
<th>BM-25</th>
<th>Language Model</th>
<th>INQUERY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top-5</td>
<td>11.95</td>
<td>10.37</td>
<td>7.74</td>
<td>10.25</td>
</tr>
<tr>
<td>Top-10</td>
<td>14.51</td>
<td>12.27</td>
<td>8.23</td>
<td>11.55</td>
</tr>
<tr>
<td>Top-50</td>
<td>16.42</td>
<td>13.36</td>
<td>9.81</td>
<td>12.43</td>
</tr>
<tr>
<td>Top-100</td>
<td>13.72</td>
<td>14.00</td>
<td>8.99</td>
<td>14.31</td>
</tr>
<tr>
<td>Top-500</td>
<td>15.77</td>
<td>14.66</td>
<td>9.05</td>
<td>14.47</td>
</tr>
<tr>
<td>Top-1,000</td>
<td>15.17</td>
<td>14.66</td>
<td>8.82</td>
<td>15.15</td>
</tr>
<tr>
<td>Top-5,000</td>
<td>15.75</td>
<td>14.32</td>
<td>8.73</td>
<td>14.43</td>
</tr>
</tbody>
</table>

FIG. 11. Verify passage size to different passage retrievers.

Four models as input to our method. For the three parameters, we choose the observed optimal settings from previous experiments (i.e., $\lambda = 0.6$, $\alpha_1 = 1.4$, $\alpha_2 = 0.5$). Figure 11 shows the experimental results with different numbers of initial retrieved passages. When feeding with exactly five initial retrieved passages, it can be viewed as the reranking improvement over the ranking model. Table 10 lists the improvement rates for different initial retrieval models.

Obviously, our method can effectively improve the results of different retrievers. In comparison to the language model,
it enhances the relatively 8.23% MRR score and 12.27% for the BM-25 models via adopting their top-10 passages. The best performance was obtained by reranking top-20 initial retrieved passages from the language model (achieving a 0.579 MRR score). It tends to converge when retrieving more than 200 passages. For this experiment, we found that the use of a large number of initial retrieved passages is not useful but also reduces time efficiency. By feeding with less than the top-1,000 passages, our method can efficiently respond to a question in less than 1 s while it becomes slow when taking tens of thousands of passages. As shown in Figure 12, there are no significant differences among the rankers when adopting more than 200 initial retrieved passages.

Conclusion

We have presented a bilingual passage-retrieval algorithm to support bilingual videoQA. Users can access Chinese videos through English or Chinese natural language questions. We also evaluated our method with large-scale use of raw videos and a question set. The experimental results showed that this system could effectively retrieve answers. Of all the tested approaches, the proposed method achieves the best performance. When porting to English, encouraging empirical results also were obtained. While the proposed method has been shown to be effective, further improvement could be obtained via correcting OCR errors. We plan to integrate speech-recognition technology to improve such errors in the future. An online demo version of the prototype system can be found at the Web site (http://140.115.112.118/bcbb/TVQS3/index.htm). Currently, we do not provide online video play (Instead, static image frames were used.) due to the speed of Internet transmission and for reasons of copyright.

References


Appendix

Optimizing Language Models

The smoothing method and its associated parameters need to be selected carefully for the language model. Typically, there are three well-known and effective smoothing methods: Jelinek-Mercer (Zhai & Lafferty, 2001), Dirichlet (Zhai & Lafferty, 2001), and two-stage language models (Zhai & Lafferty, 2002). By following Zhai and Lafferty (2001), we verify $\lambda$ for the Jelinek-Mercer method with 0.05, 0.15, ..., 0.95, and the Dirichlet prior $\mu$ with 20 equal parts. Next, the optimal and observed $\mu$ is used to verify the two-stage language models through the interpolation parameter ($\lambda'$). For a fair comparison, we use the same validation dataset (150 questions) to tune the parameters. According to Table A1, the best QA performance is obtained by adopting the two-stage smoothing method with $\mu = 200$ and $\lambda' = 0.15$. Table A1 shows the detailed results of the different smoothing strategies and parameters.

<table>
<thead>
<tr>
<th>$\lambda$</th>
<th>MRR</th>
<th>Prior ($\mu$)</th>
<th>MRR</th>
<th>$\mu$, $\lambda'$</th>
<th>MRR</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td>0.536</td>
<td>10</td>
<td>0.541</td>
<td>200, 0.15</td>
<td>0.565</td>
</tr>
<tr>
<td>0.15</td>
<td>0.540</td>
<td>250</td>
<td>0.564</td>
<td>150, 0.35</td>
<td>0.564</td>
</tr>
<tr>
<td>0.25</td>
<td>0.547</td>
<td>500</td>
<td>0.535</td>
<td>200, 0.25</td>
<td>0.564</td>
</tr>
<tr>
<td>0.35</td>
<td>0.548</td>
<td>750</td>
<td>0.534</td>
<td>250, 0.05</td>
<td>0.564</td>
</tr>
<tr>
<td>0.45</td>
<td>0.551</td>
<td>1000</td>
<td>0.522</td>
<td>100, 0.45</td>
<td>0.563</td>
</tr>
<tr>
<td>0.55</td>
<td>0.553</td>
<td>1250</td>
<td>0.510</td>
<td>150, 0.25</td>
<td>0.562</td>
</tr>
<tr>
<td>0.65</td>
<td>0.545</td>
<td>1500</td>
<td>0.504</td>
<td>150, 0.15</td>
<td>0.560</td>
</tr>
<tr>
<td>0.75</td>
<td>0.537</td>
<td>1750</td>
<td>0.493</td>
<td>100, 0.55</td>
<td>0.560</td>
</tr>
<tr>
<td>0.85</td>
<td>0.544</td>
<td>2000</td>
<td>0.485</td>
<td>250, 0.25</td>
<td>0.559</td>
</tr>
<tr>
<td>0.95</td>
<td>0.498</td>
<td>2250</td>
<td>0.483</td>
<td>50, 0.65</td>
<td>0.559</td>
</tr>
</tbody>
</table>

$\text{MRR} = \text{main reciprocal rank score.}$

TABLE A1. Parameter settings for the Language Model.

American Chapter of the Association for Computational Linguistics, Rochester, NY.


TABLE A1. Parameter settings for the Language Model.