Towards Content-Based Relevance Ranking for Video Search

Wei Lai Xian-Sheng Hua Wei-Ying Ma
Microsoft Research Asia
No.49, Zhichun Road, Haidian District, Beijing, P.R.China
{weilai, xshua, wyma}@microsoft.com

ABSTRACT

Most existing web video search engines index videos by file names, URLs, and surrounding texts. These types of video metadata roughly describe the whole video in an abstract level without taking the rich content, such as semantic content descriptions and speech within the video, into consideration. Therefore the relevance ranking of the video search results is not satisfactory as the details of video contents are ignored. In this paper we propose a novel relevance ranking approach for Web-based video search using both video metadata and the rich content contained in the videos. To leverage real content into ranking, the videos are segmented into shots, which are smaller and more semantic-meaningful retrievable units, and then more detailed information of video content such as semantic descriptions and speech of each shots are used to improve the retrieval and ranking performance. With video metadata and content information of shots, we developed an integrated ranking approach, which achieves improved ranking performance. We also introduce machine learning into the ranking system, and compare them with IR–model (information retrieval model) based method. The evaluation results demonstrate the effectiveness of the proposed ranking methods.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval - *Search process*

General Terms: Algorithms, Performance, Experimentation.

Keywords

Video search, Relevance ranking, Content-based ranking

1. INTRODUCTION

Multimedia search has become an active research field due to the rapid increase of online-available content and new practical applications. Search technology is considered the key to navigating the Internet's growing media (video, audio and image) collections. Google Yahoo, Blinkx and other search companies have provided elementary video search engines. However, existing video search engines are all based on the text information related to the video which can be retrieved from web pages, such as file names, URLs, and surrounding texts. These types of textual information can be considered as "metadata" of the video since they only roughly describe the video. There is no doubt that text searching is the most efficient way to retrieve information (even when searching for videos), because it well matches the manner of human thinking. However, only using metadata is far form people's expectation in video searching, because even the best case scenario, the metadata is only the highly concentrated overview of a video, with many losses on details.

In general, a video consists of many shots and sub-events with a temporal main thread. The video should be segmented into smaller retrievable units that are directly related to what users perceive as meaningful. Much research has concentrated on segmenting video streams into "shots" using low level visual features [1]. Each segment has its own scenes and meanings. In many cases, when users query a video, they intend to find some desired clips in the video instead of viewing it thoroughly. However, this can seldom be achieved by searching the surrounding text which is related to the whole video.

Much content information can be used to search videos and shots. In content-based video retrieval systems, video shots can be classified into or annotated by several semantic concepts. The most substantial works in this field are presented in the TREC Video Retrieval Evaluation (TRECVID) community [2]. In addition, speech is also significant information which has close connection to video contents. Some videos are associated with transcripts/closed captions which are provided by content provider. Using ASR (automatically speech recognition) to generate speech text is another practical solution.

In this paper, with the video metadata and content information of video shots, we index and rank the videos in a way similar to general text-based web page search engines. The IR-model, which is widely employed in text information retrieval and web page search, will be applied to rank the search results by examining relevance between query and indexed information (including both metadata and content information). To fully utilize the content information and get a better ranking performance, we integrate the "shot relevance" into "video relevance". That is, the ranking is decided not only by the relevance of video (metadata of the entire video), but also by all the relevant shots within the video.

We also apply learning based method to rank the search results based on a set of features extracted from the corresponding query, video metadata, and content information.

The rest of this paper is organized as follows. Section 2 introduces the IR-model based ranking, including extraction of video metadata and content information, and a ranking method integrating these two types of information. In section 3, a learning based ranking approach is presented. Section 4 compares the ranking performance evolution results, and Section 5 concludes the paper.

MM'06, October 23?7, 2006, Santa Barbara, California, USA.
ACM 1-59593-447-2/06/0010.

2. IR-MODEL BASED RANKING

2.1 Relevance Evaluation of Text Search

In the traditional text retrieval and web page search, IR (Information Retrieval) models are usually used to evaluate the relevance between a query and a document. BM25 [3] is one of the frequently used evaluation methods. Given a query *Q* and a document *D*, the relevance between *Q* and *D* can be modeled as the summation of the relevance between each query term (word) *t* in *Q* and *D*:

$$R(D,Q) = \sum_{t \in Q} R(D,t) \tag{1}$$

where:

$$R(D,t) = \frac{(k_1+1) \cdot tf(t,D)}{k_1 \cdot ((1-b) + b \cdot \frac{|D|}{avdl}) + tf(t,D)} \cdot w(t) \tag{2}$$

$$w(t) = \log \frac{N - df(t) + 0.5}{df(t)} \tag{3}$$

Here k_1 and *b* are parameters. *tf*(*t*,*D*) is term frequency, means the frequency of term *t* appears in document *D*. *df*(*t*) is document frequency, means the frequency of document which contains term *t* within all documents. |*D*| stands for the length of document *D*.

The basic idea of this IR model can be explained as, if the query term appears in document more frequently (higher *tf*), and the query term is more unique that less documents contain it (lower *df*), the query will be more relevant to the document.

2.2 Index and Rank the Video Information

The video data used in our experimental system are from MSN Video (http://video.msn.com/), which contains 7230 videos.

2.2.1 Metadata of the video

Because the videos in our data set are made by professional content provider, there is rich meta information that describes each entire video with brief text. Each video has the following metadata fields: *headline*, *caption*, *keywords*, *source*, *video URL*, *thumbnail image URL*, *related web page anchor text*, *page URL*, etc. Besides these types of textual information, some format information of the video, such as video length, frame size, bit rate, and creation date are also extracted. Some selected information fields of video metadata are listed in Table 1.

Table 1. Video metadata

Field	**Example Value**
Headline	Discovery launches
Caption	July 26: Watch the entire launch of **space shuttle** D…
Source	MSNBC
Keywords	Technology, science, **Space**, Partner Codes …
Video URL	http://www.msnbc.msn.com/default.cdnx/id/871313…
Link anchor	MSNBC.com's Technology and Science front
Link URL	http://www.msnbc.msn.com/id/3032118
date	7/26/2005 4:40:48 PM
video length	609.72 seconds
Frame size	320 x 240
Bit rate	180 Kbps

For the videos contained in general web pages, some attributes mentioned above may not be obtained directly, but the surrounding texts, URL, filename can be extracted as the metadata fields of the video.

These information fields correspond to document *D* in Section 2.1. Different fields can be represented by different type of *D* (D_i in Equation 4). The overall relevance can be calculated by the weighted summation of the relevance of all fields. The weight of the fields (DW_i in Equation 4) can be determined by their importance, significance, and representativeness to the video.

$$R(Video,Q) = \sum DW_i \cdot R(D_i,Q) \tag{4}$$

In our system, four major information fields from video metadata are selected to be indexed: *headline*, *caption*, *keywords*, and *source*. *Headline* is a highly representative description of the video content. *Keywords* are also good recapitulative terms. For these two fields, higher weights are set. *Caption* is a more general and detail depicts for the video; *Source* provided a higher level and less relevant information, they will be set lower weights for ranking. Table 2 gives out the weights of fields in our experimental system.

Table 2. Weights for relevance evaluation

Fields	**weight**
Headline	10
Keywords	10
Caption	5
source	1

2.2.2 Content information of the video shots

There is plenty of information in the visual/audio content of the video sequence, which can not be sufficiently presented by the aforementioned textual video metadata. We can build a set of models that can be applied to automatically detect a corresponding set of concepts such that each video shot can be annotated with a detection confidence score for each concept. Successful concept modeling and detection approaches have been introduced in TRECVID, relying predominantly on visual/aural analysis and statistical machine learning methods [4]. The LSCOM-lite Lexicon [5] designed for the TRECVID 2005 Benchmark consists of more than 40 concepts spread across multiple concept-types such as object, events, site etc. Though the size of the lexicon is still far from practical application for general Web-based video search, this semantic information is promising to enable real content-based video search, and therefore it is applied in our ranking system.

Besides visual contents, information from audio channel, especially the speech, is also very useful for searching videos.. In our experimental system, we use Microsoft speech recognition engine (with a large vocabulary). This engine gives recognized words with a start timestamp, length, and a recognition confidence value, which are very useful for later indexing and ranking. The speech texts are allotted and assigned into video shots, according to the timestamp of words and video shots.

The content information is associated with individual video shot, which consist of semantic keywords (with corresponding detection confidences), and speech words (with recognition confidences). The confidences of words will act as weights of term frequencies *tf* to calculate the relevance in Equation (2).

2.3 Integrated Ranking with Metadata and Content Information

To combine metadata and content to rank the videos, we index the videos by metadata and index the video shots by content information separately, and then integrated these two rank lists, named *video list* and *shot list*, to form a final ranking. The integrated ranking returns search result by video, but taking all the relevance shots within this video into consideration.

For *video list*, each item is a video. Let $item_i^v.vid$ denotes the video ID of the i^{th} item, $item_i^v.score$ denotes the ranking score of the i^{th} item. For *shot list*, each item is a shot from a video. Let $item_i^s.vid$, $item_i^s.sid$, $item_i^s.score$ donote the video ID which the shot belong to, the shot ID within the video, and the ranking score of the i^{th} item respectively.

The integrating process is presented in Algorithm 1. The basic idea is that, all the ranking score of the relevance shots within the video are accumulated to the ranking score of the video, with corresponding weights. The relevant shots in the video will be highlight when displaying the video as search result.

```
new a integrated result list (item denotes as $item_i^I$)
for each $item_i^v$ in video list{
    new $item_i^c$;
    $item_i^I.vid = item_i^s.vid$;
    $item_i^I\ score = item_i^v.\ score * Weight\_v$;
    for each item $item_i^s$ in shot list{
        if($item_i^v.vid == item_i^s.vid$) {
            $item_i^I\ addshot(item_i^s.sid)$;
            $item_i^I\ score \mathrel{+}= item_i^s.\ score * Weight\_s$;
            remove $item_i^s$ from shot list
        }
    }
    remove $item_i^v$ from video list
    add $item_i^I$ to integrated list
}
add the remaining video list and shot list into the integrated list
sort the integrated list by $item_i^I.score$.

// $Weight\_v$ and $Weight\_s$ are weights for score accumulating
```

Algorithm 1. Generate integrated rank list.

3. LEARNING BASED RANKING

3.1 Extracted Features

IR-model based ranking just consider some basic features such as term frequency *tf*, document frequency *df*, and document length, etc. In the learning based approach, more features are extracted from the query, metadata, and content information. To be clear, suppose the query contains three terms "a b c", we compute the following features from each document field:

Ordered match: the frequency that both "a" and "b" appeared in the indexed text, and "b" appears after "a".

Partly exact match: the frequency that "a b" or "b c" appeared in the indexed text.

Exact match: the frequency that "a b c" appeared in the indexed text.

Query length: number of query terms

For the content information, each word has a confidence value, we also consider:

Weighted *tf*: Term frequency with confidence weighted,

High confident match: query term match with words with high confidence.

High confident words: words with high confidence in the indexed text.

Some non-textual, query-independent features, such as shot length, video length, frame size, bit rate, etc, are also taken into account.

By counting in the combinations of several document fields and query terms (or part of query), we have about 50 dimensional features in total for a query and search result to form a sample.. The GroundTruth of sample is the relevance judgments between a query and a result, which is collected by a user labeling system introduced in the next section.

3.2 Neutral Network based Ranking

Traditionally the learning algorithms are used for classification problems. For ranking problems, one possible way is organize the samples by pairs. Pair sample (x_1, x_2) are considered as a positive sample, if x_1 are ranked ahead of x_2, and vice versa. The loss function is also formulated in pair wise. RankNET [6] is used in our implementation to train the ranking model and to validate the performance.

About half of the labeled data are used in training, and the second half are used for validation.

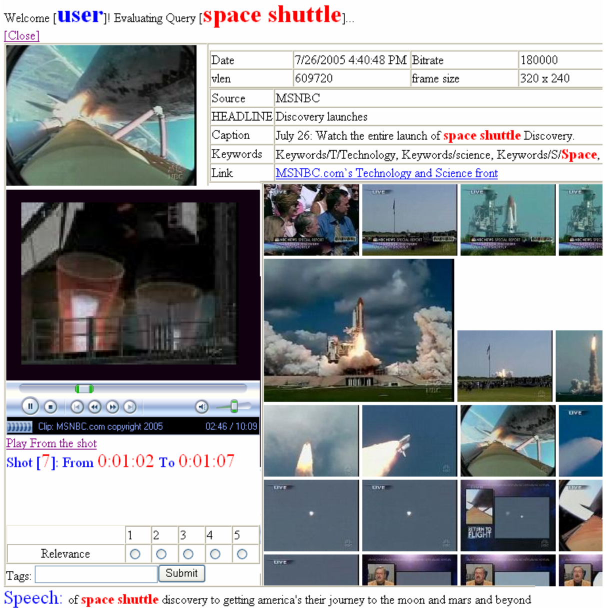
4. EVALUATIONS

4.1 Data Preparation

To evaluate the ranking performance of our proposed methods, we developed a user labeling tool to collect some query-result relevance judgments.

The video data set we used in our experiment includes news video, TV programs, movie trailers, advertisements, etc. According to the characteristics of the content of these videos, we selected some news related queries, such as hot events, hot place, and hot person names, to evaluate the ranking performance. For each query, we use the IR-model based ranking describe in Section 2 to generate a result list, and randomly select some results form the list to label. Considering the labeling workload, for each labeler and each query, 9 results are select from the list. To make the selected query-result samples have a good uniformity on distribution, 3 results are randomly selected from the first 1/3 part of the list, 3 are from the second 1/3 part, and the other 3 are from the last 1/3 part. The order of these 9 selected results is shuffled and then provided to users to do relevance labeling.

In the labeling tool, for a query and a result, user can see all the information of the result, including the file format information (frame size, bit rate, video length, etc), description (headline, caption, keywords), video thumbnail, video (in a video player), thumbnails of video shots, the speech text of the relevant shots. The words matched with query terms are highlighted. See Figure 1. If there are relevant shots in the result, the thumbnails of them are displayed with doubled size. The shot number, time information, and the speech are also shown in the interface. Users are asked to read the displayed information, browse the thumbnails, and play the video and shots (a tool button is provided to play from one shot) to give a relevance judgment from 1, 2, 3, 4, and 5, which represent *bad, fair*, *good, excellent,* and *perfect*, respectively.



Welcome [**user**]! Evaluating Query [**space shuttle**]...

[Close]

Date	7/26/2005 4:40:48 PM	Bitrate	180000
vlen	609720	frame size	320 x 240

Source	MSNBC
HEADLINE	Discovery launches
Caption	July 26: Watch the entire launch of **space shuttle** Discovery.
Keywords	Keywords/T/Technology, Keywords/science, Keywords/S/**Space**,
Link	MSNBC.com's Technology and Science front

Clip: MSNBC.com copyright 2005 02:46 / 10:09

Play From the shot

Shot [7]: From 0:01:02 To 0:01:07

	1	2	3	4	5
Relevance	○	○	○	○	○

Tags:

Submit

Speech: of **space shuttle** discovery to getting america's their journey to the moon and mars and beyond

Figure 1. Relevance labeling tool

In our experiment, ten users are invited to do labeling, and about 2,000 relevance judgments of query-result samples are collected.

4.2 Precision Performance of Ranking

We have conducted a comparison between the 4 approaches listed below:

MR: Ranking only based on video metadata (Section 2.2.1).

CR: Ranking only by content information (Section 2.2.2).

RI: Integrated Ranking described in Section 2.3

RN: RankNET based ranking described in Section 3.2.

The precision in top *N* of the rank lists of all the labeled queries is used to evaluate the performance of ranking method.

$$Precision@N = \frac{relevant\ \ labeled\ \ results\ \ in\ top N}{total\ \ labeled\ \ results\ \ in\ top\ N} \qquad (5)$$

In our implementation, the judgment *Perfect* or *Excellent* are considered as relevant results, while other judgments are treated as irrelevant results. The *Presicion@N* (*N*=1 to 5) of the 4 ranking methods are shown in Table 3.

From the results, we can see that:

1) Precisions of ***MR*** are very low. Only using video metadata will result in a poor performance, since details of the video content are ignored. The content information is more effective to search and rank video than metadata, as precisions of ***CR*** are higher than that of ***MR***.

2) Precisions of ***RI*** are much higher than that of ***MR*** and ***CR***. By combining video metadata and content information, the performance is significantly improved and reaches an acceptable level, which shows that content-based relevance ranking is a promising approach.

3) ***RN*** has a good performance, even better than ***RI***. Comparing to IR-model based ranking, more features are included to learning the relevance. The result implies that the learning method can organize the information for ranking in a more effective way.

Table 3. Precision of the ranking approaches

Precision@?	1	2	3	4	5
MR	0.305	0.326	0.299	0.259	0.228
CR	0.544	0.526	0.571	0.550	0.522
RI	0.796	0.727	0.684	0.634	0.606
RN	0.805	0.746	0.763	0.717	0.669

5. DISCUSSIONS AND CONCLUSION

We have presented a novel content-based approach to rank video search results. In addition to the video metadata, more detailed content information in the video is used to improve the relevance ranking of video search results. The videos are segmented into shots, which can carry rich content information such as semantic concept keywords and speech. With the video metadata and content information, we proposed an IR-model based ranking method and a learning-based ranking method. Evaluation of the top ranked results shows that the proposed ranking methods have significantly improved performance comparing to the approach use video metadata only, which is frequently used in existing web video search engines.

In future work, more types of content information can be integrated into our ranking scheme, such as content-based quality metric, user comments and rating for videos shared in web communities. Moreover, how to define effective semantic concepts, i.e., video semantic ontology, that facilitate video searching and ranking is also a challenging problem., which is also one of our future works.

6. REFERENCES

[1] Hong-Jiang Zhang, A. Kankanhalli, and S. Smoliar, "Automatic Partitioning of Full-motion Video," A Guided Tour of Multimedia Systems and Applications, IEEE Computer Society Press, 1995.

[2] http://www-nlpir.nist.gov/projects/trecvid

[3] S. E. Robertson, S. Walker, and M. Beaulieu. Okapi at TREC–7: automatic ad hoc, filtering, VLC and filtering tracks. In Proceedings of TREC'99.

[4] M. Naphade, J.R. Smith, F. Souvannavong, "On the Detection of Semantic Concepts at TRECVID," ACM Multimedia, ACM Press, New York, NY, pp. 660-667, Oct. 10-16, 2004

[5] M. Naphade, L. Kennedy, J.R. Kender, S.F. Chang, J.R. Smith, P. Over, A. Hauptmann, "LSCOM-lite: A Light Scale Concept Ontology for Multimedia Understanding for TRECVID 2005," IBM Research Tech. Report, RC23612 (W0505-104), May, 2005.

[6] Chris Burges, *et.al*, "Learning to Rank using Gradient Descent", ICML 2005, Bonn, Germany, pp.89-96, August 7-11, 2005.