FINDING SPECIFICATION PAGES ACCORDING TO ATTRIBUTES

Naoki Yoshinaga: *Japan Society for the Promotion of Science*
Kentaro Torisawa: *Japan Advanced Institute of Science and Technology*

1. Introduction

Our Target: Specification Page

A page that provides information on the object's *comprehensive set of attributes* in *well-formatted style*

label what we want to know about the objects

Ex. Query `Actress, Audrey Hepburn'

Specification page

Audrey Hepburn:
She was born in …
Personal data:
Name: *Edda Van Heemstra Hepburn-Ruston*
Birthdate: *May 4, 1929*
Sex: *Female*
Partner: N/A
Bio: …

We can grasp information on several attributes *at a glance*

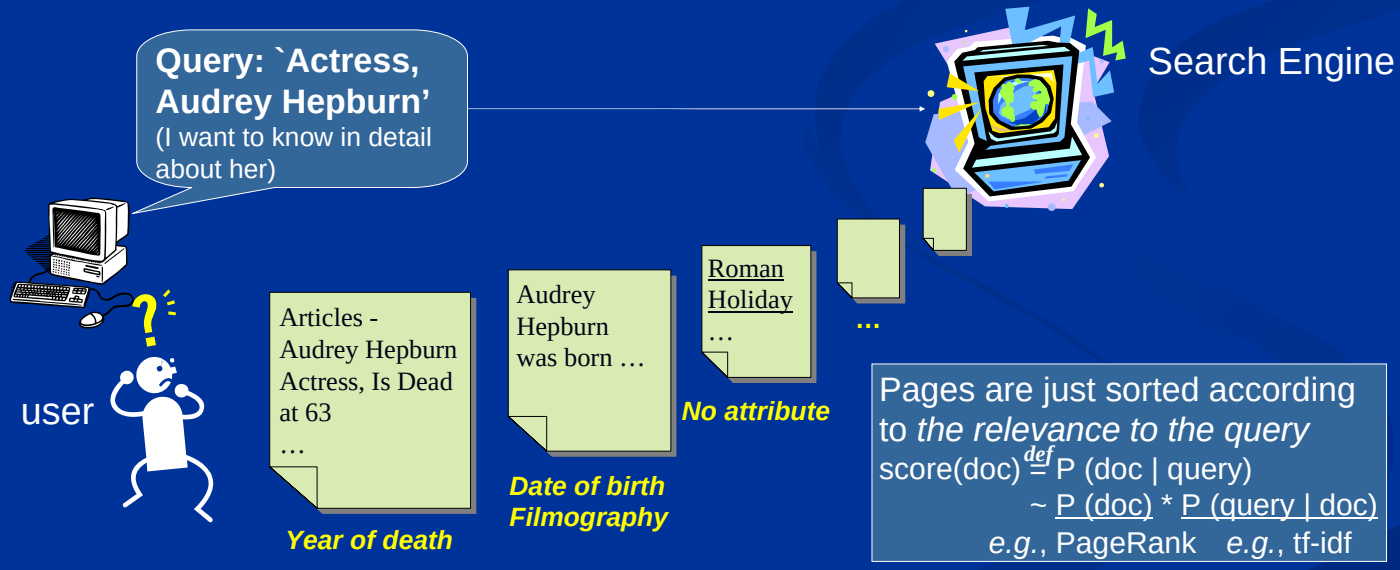
Search engine' top

Articles --- Audrey Hepburn, Dead at 63
Audrey Hepburn, the actress who epitomized Hollywood chic in the 1950's and 60's, died yesterday at her home in Tolochenaz, near Lausanne, Switzerland.

We can find only her year and place of death

Research Background

Encyclopedic web search, to acquire detailed information about an object from the web, has been recently quite popular; however
- Query to database (wikipedia): limited coverage for both domain and instance
- Query to a normal search engine: need to wade through several pages to excavate pieces of information scattered in those pages



Pages are just sorted according to *the relevance to the query*
score(doc) $\stackrel{def}{=}$ P (doc | query)
~ P (doc) * P (query | doc)
e.g., PageRank *e.g.*, tf-idf

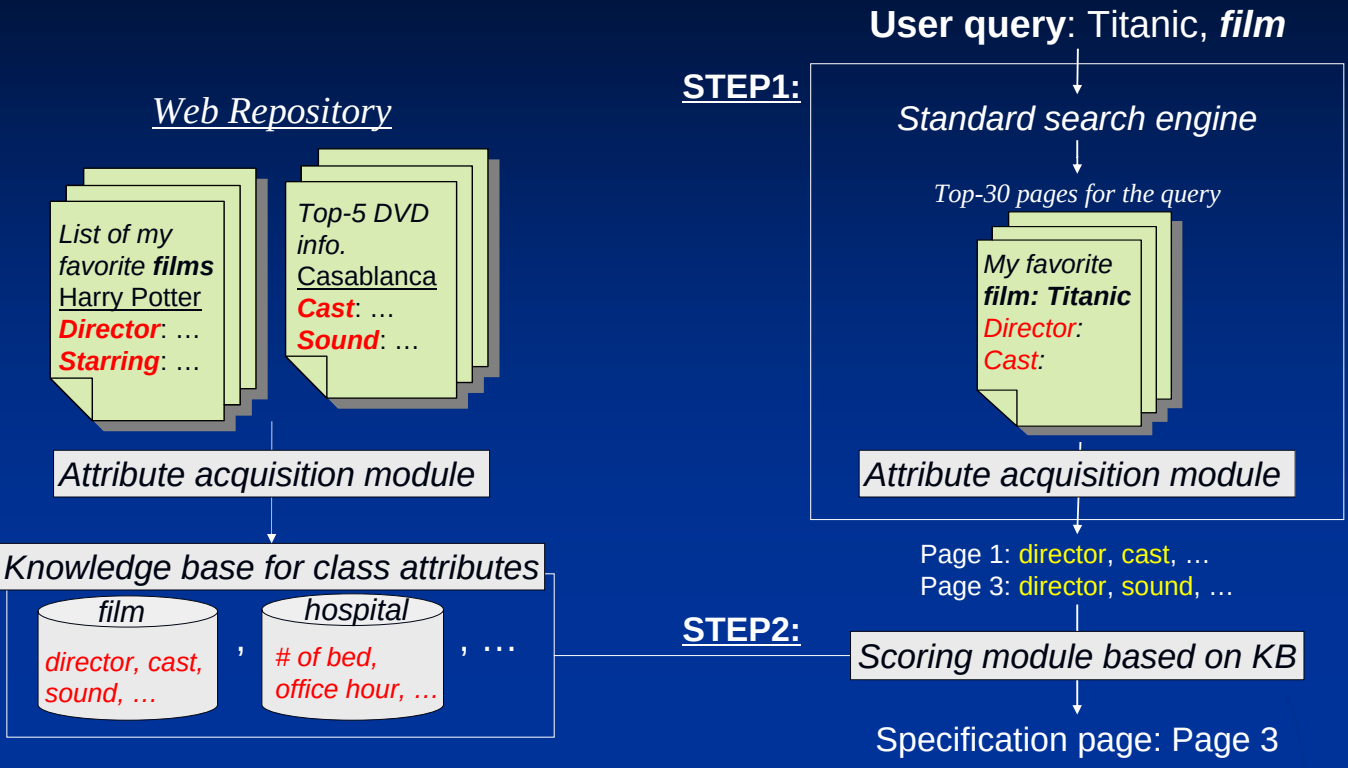
Common wisdom: augment a query with *attributes of the object*

Some attributes are often paraphrased (recall deterioration); others are hard to verbalize for a novice user

If we find a representative specification page for an object, we can **rapidly** obtain information we want to know about the object !

2. Approach

We rank pages that include an object (*e.g.*, Titanic) according to *attributes of classes (`film')*, *off-line* extracted from web pages that describe the classes



Construction of Knowledge Base (KB) for Class Attributes

1. Collect web pages that describe the target class as a topic
   - pages which have the class name surrounded by some HTML tags (*e.g.*, 'title')
2. Extract candidate attributes by *layout and symbolic decoration cues*
   - only expressions that pass through morphological-analysis and stop-words filters

HTML description

```
The following lists my favorite films.<BR />
<H2>Titanic (year: 1997)</H2>
Director/ James Cameron
The details:
<TABLE><TR><TD>Genre</TD><TD>Romance</TD>
</TR><TR><TD>Runtime</TD><TD>194min.</TD>
</TR><TR><TD>..</TD><TD>..</TD></TR></TABLE>
```

Browser view

The following lists my favorite films.
Titanic (year: 1997)
Director/ James Cameron
The details:

Genre	Romance
Runtime	194 min.
..	..

3. Filter candidate attributes by ***site frequency,*** the number of *websites* where the attributes are extracted (~ the number of *authors* who used the attributes)

   Website for a page was defined by digging through its URL until the directory included a file whose name matched `/^(?:index|default|main)¥..+/`

Specification Page Finding

STEP 1: Extract *page attributes*, A_p, from each page, p, for the target object

STEP 2. Score each page according to page attributes, A_p, and class attributes, A_C

$$score(p) = \frac{\#(A_p \cap A_C) \times ratio(A_p, A_C)}{ave(A_p, p) \times text\_size(x, p)}$$

$ratio(A_p, A_C)$: the percentage of the page attributes that are included in the class attributes

$ave(A_p, p)$: the average number of appearances of attributes on the page

$text\_size(x, p)$: the length of the text enclosed by HTML tags that first contains the object name

3. Preliminary Evaluation

Experimental Settings:

- Local Japanese web repository (*local_web*, 0.7 TB incl. tags)
- KB: 30 attributes for each class (built from 10K pages for each class, accuracy: 79%)
- Three systems output a single page for 100 objects in 10 classes
  - **Google**: output a page that is ranked top by Google
  - **SP**: find a page with KB by re-ranking Google's top 30 outputs
  - **SP***: find a page with KB from 10K pages randomly chosen from local_web

Evaluation Scheme:

1. 3 subjects determine *4* attributes they want to associate with objects of each class
2. examine whether pages outputted by the three systems refer to the object
3. score each page by the number of attribute-value pairs (0-4) that the page contains

Experimental results

Class Name	# objects	Google	SP	SP*
Paperback	7/10	***3.33**	2.67	*1.95
Racehorse	6/10	***4.00**	***4.00**	*3.33
Actress/Actor	4/10	***1.58**	1.50	1.00
Digital camera	4/10	*1.50	***3.08**	1.33
Baseball Player	1/10	0.33	**1.33**	0.33
Hospital	4/10	0.33	***3.33**	1.25
Museum	6/10	0.28	*3.00	***3.33**
Amusement Park	5/10	1.07	*2.80	***3.13**
Wine	5/10	*1.53	1.53	***2.73**
Corporation	0/10	NA	NA	NA
Weighted Mean	42/100	1.81	**2.75**	2.33
Best-5 (*)		2.59	**3.26**	2.86

Authoritative websites had a database for the class

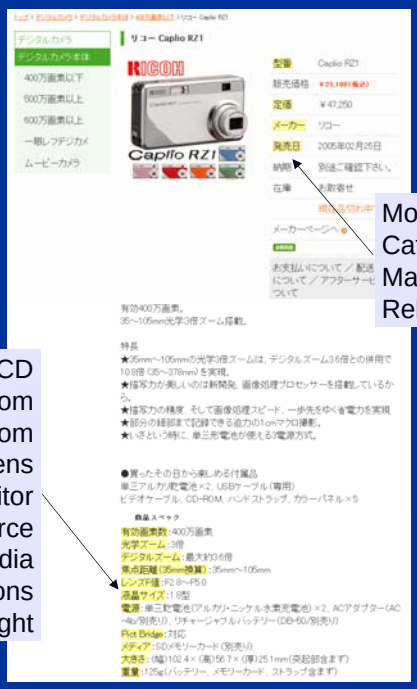
A good specification page was included in Google's top 30

A good specification page wasn't included in Google's top 30

Findings:

- Google's ranking was useful to chose pages that described an object as a topic
- Combination of Google's ranking and our attribute-based scoring was the best
- SP took just 1 sec more than Google for re-ranking

SP output for Caplio RZ1 (13 attributes)



Google's top page for Caplio RZ1(1 attribute)