Automated Tagging to Enable Fine-Grained Browsing of Lecture Videos

K.Vijaya Kumar
(09305081)

under the guidance of

Prof. Sridhar Iyer

June 28, 2011

Outline

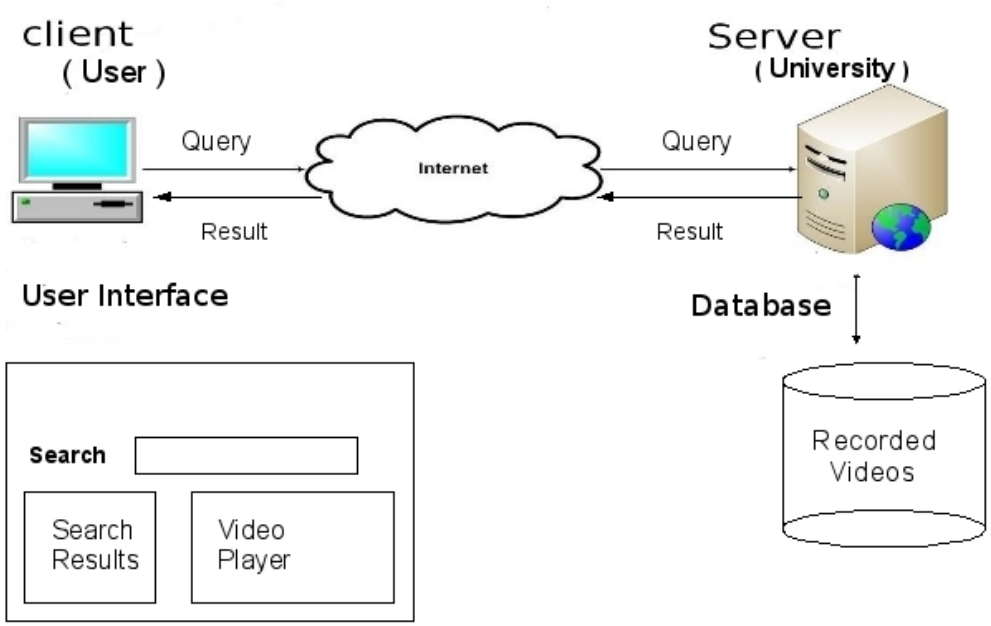
Outline

Introduction

- Lecture video recordings are widely used in distance learning
- To make best use of the available videos a system called **Browsing System** is required
- Purpose of the browsing system is to provide search facility in the lecture video repository
- **Problem Statement** :
  To develop a browsing system which is useful for users to find their required video content easily

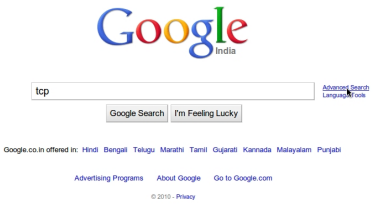
Video Browsing System

- It takes keywords from users and gives them lecture videos matching their keywords

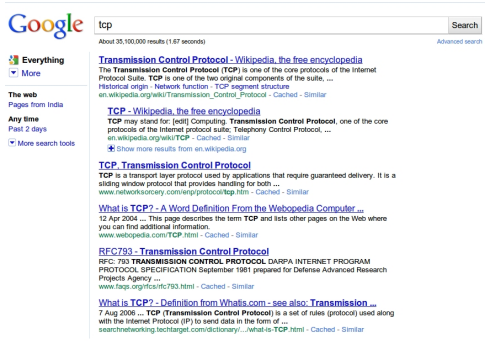


Outline

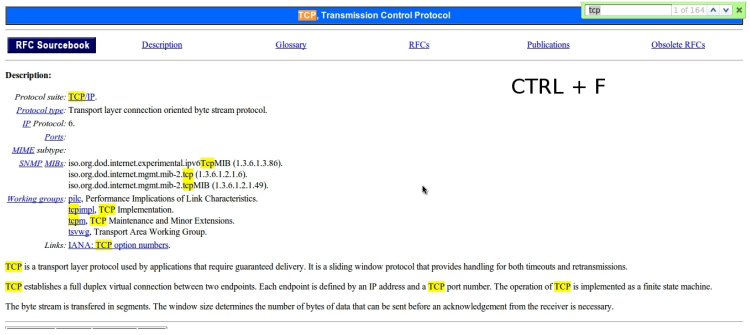
Text Search Example



(a) Query



(b) Results



(c) Finding Info

Figure: Google Search

Can we do the same in Lecture Videos ?

- Yes, We can provide the same type of search facility in lecture videos based on their contents

Example Scenarios

- Portion of video where *Matrix Multiplication* is discussed in a programming course lecture
- Searching for a video which discusses *Quick Sort* in a Data Structures course videos
- Finding video results containing *Double Hashing* in lecture video repository

Techniques for Searching in Lecture Videos

- **Meta data based** :
  Uses data such as video title, description or comments associated with the video
- **Content based** :
  Based on data extracted from lecture videos, which represents contents present within it

How You Tube Searches Videos?

- Youtube video search is based on meta data associated with videos
- Meta data include video title, description and tags

Outline

Example Lecture Video Repositories

- CDEEP[5] : No search feature
- NPTEL[16] : No search feature
- freelecturevideos.com[8]
- videolectures.net[20]
- Lecture Browser, MIT[13]

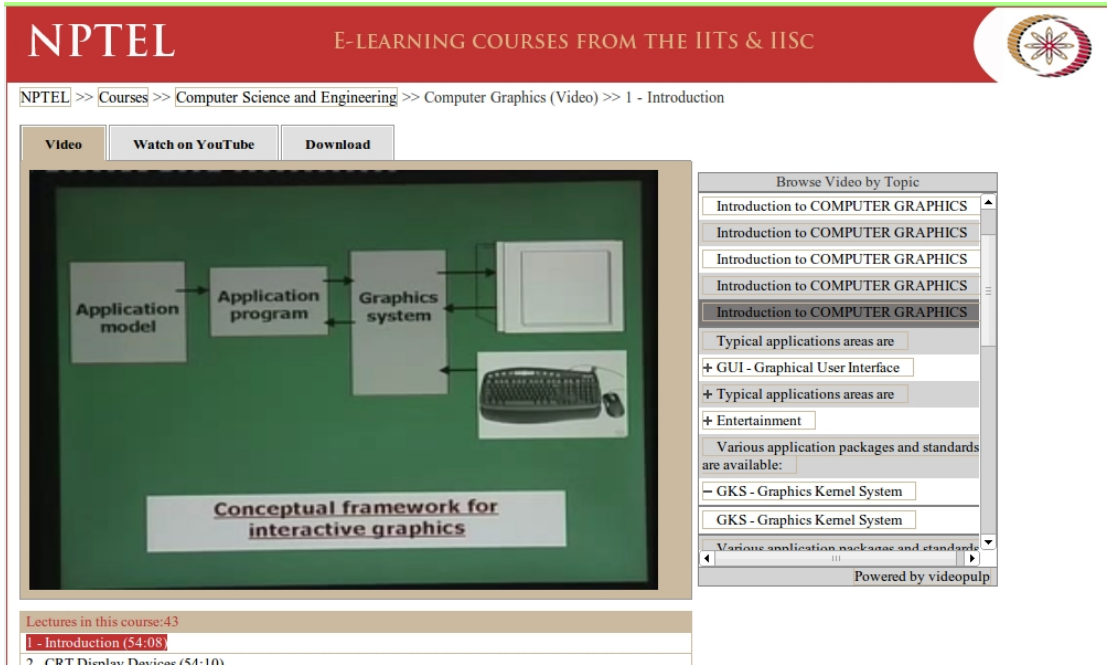
Some more

- Academic Earth[1]
- Youtube Edu[23]

Link to list of available educational video repositories is at[15]

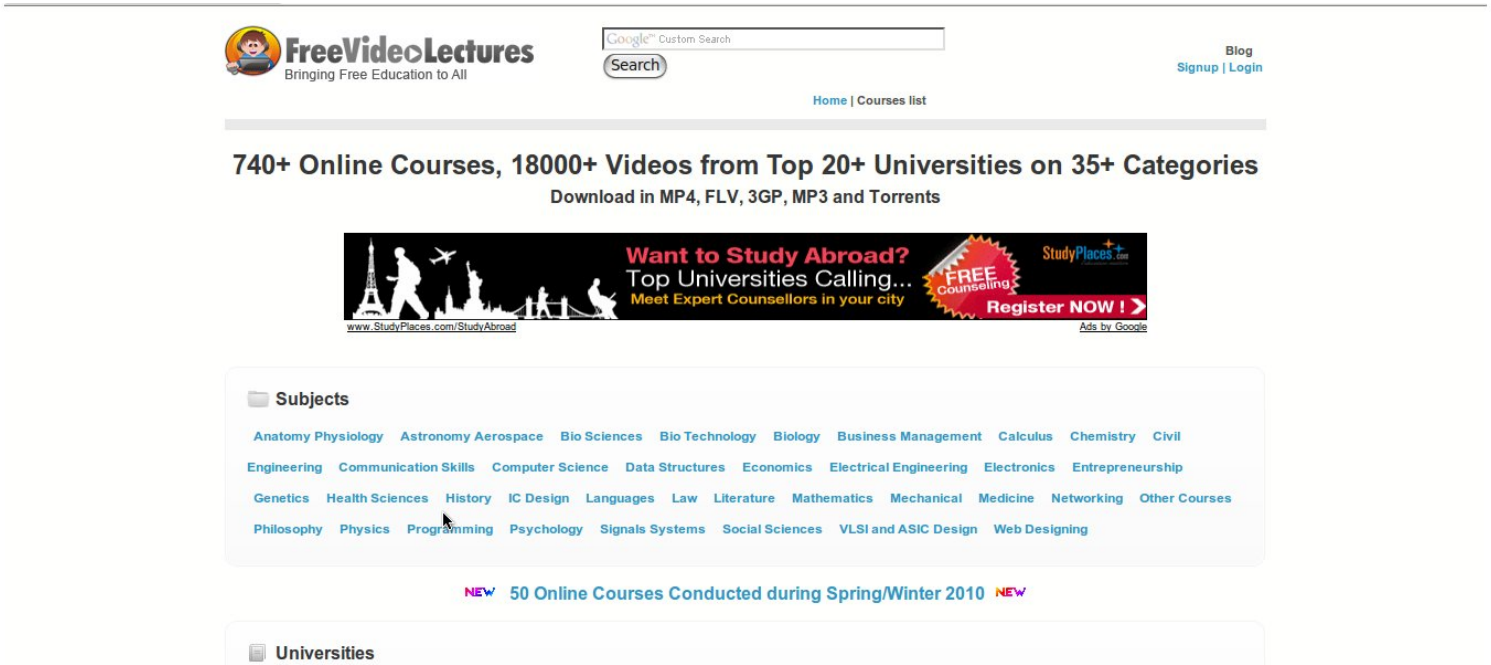
Slide Index feature in NPTEL

- Recently launched
- Through a video processing company called videopulp [21]



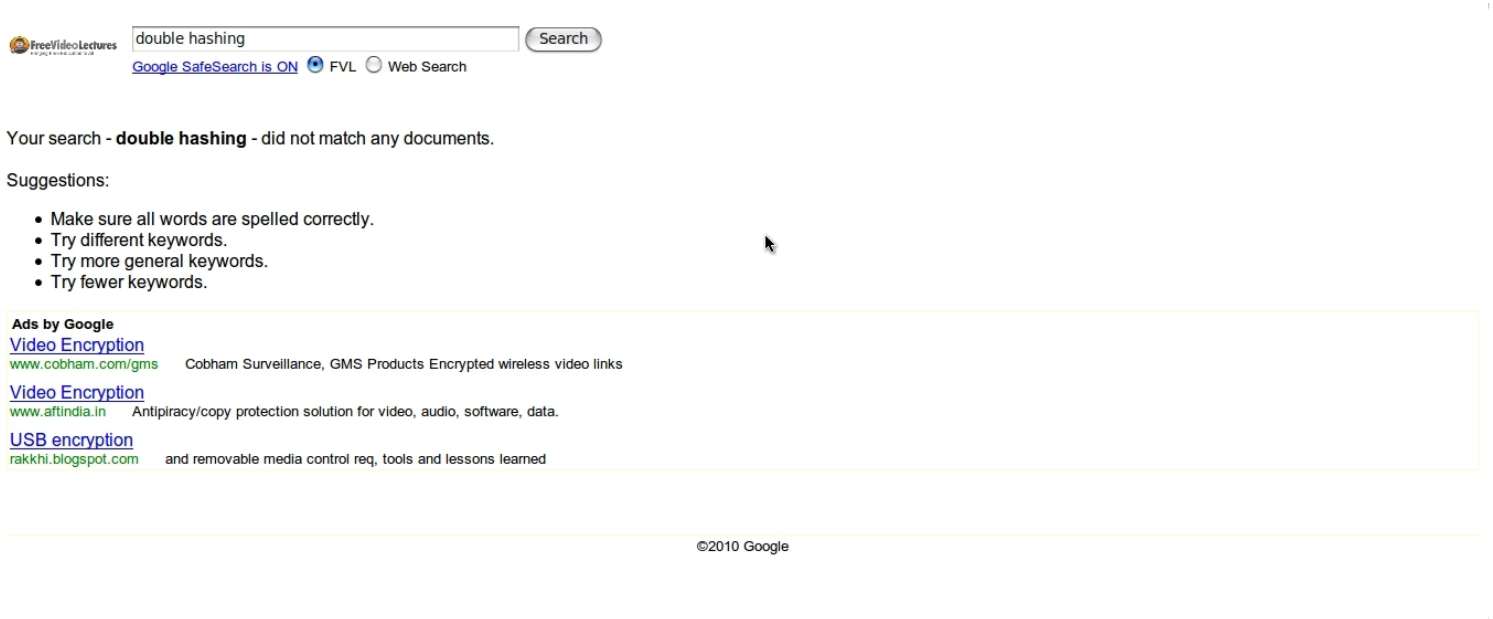
freevideolectures.com

- Provides Google custom search to index textual data
- Topic Looked for : **Double Hashing**



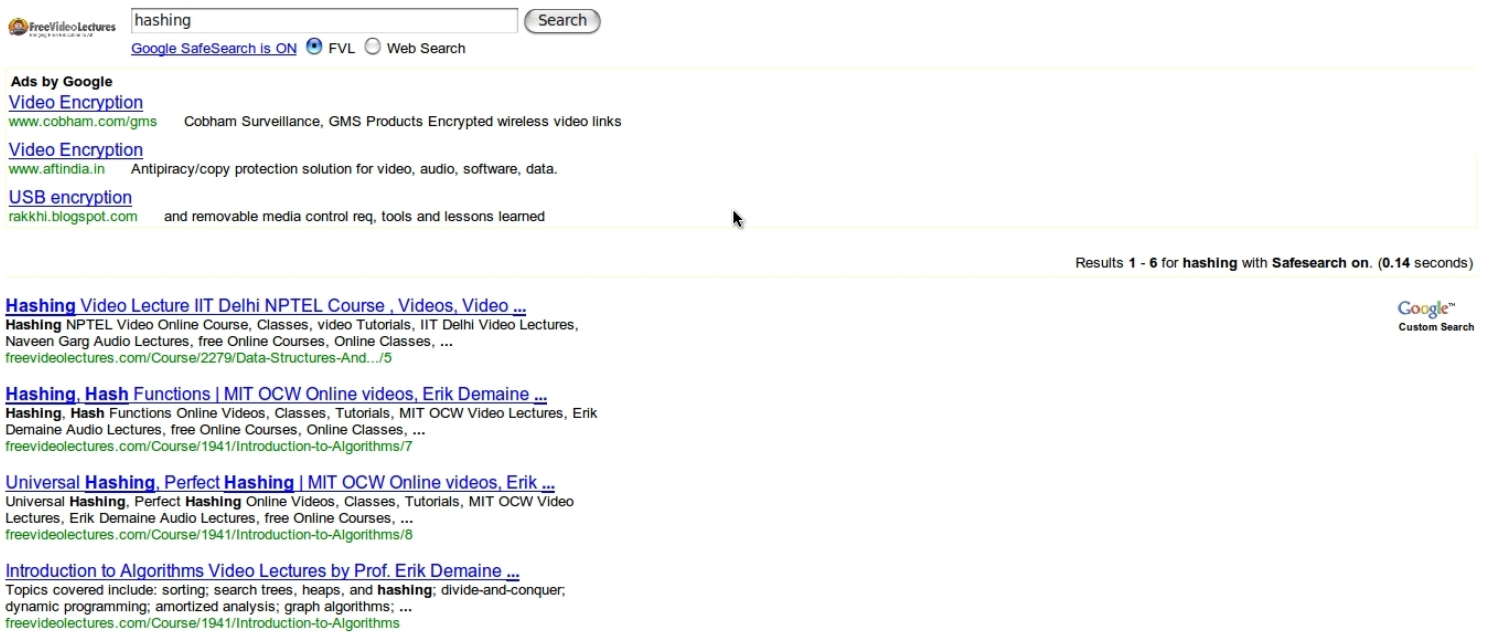
freevideolectures.com

- Keyword : **double hashing**
- Result : Your search - double hashing - **did not match any documents**.



freevideolectures.com

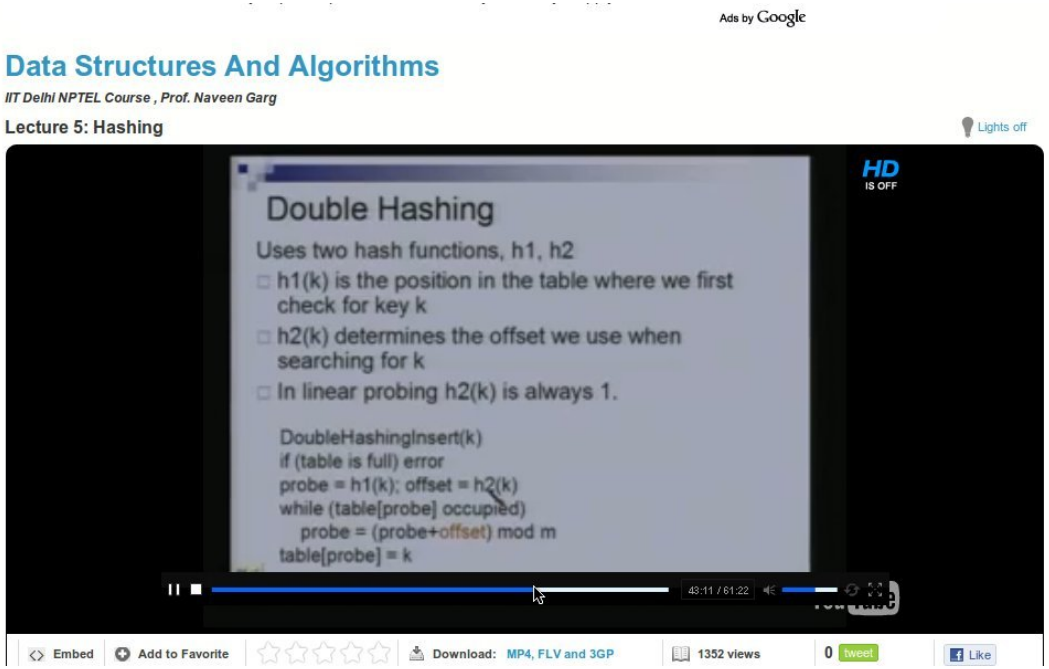
- Keyword : hashing
- Result : 6 video results



freevideolectures.com

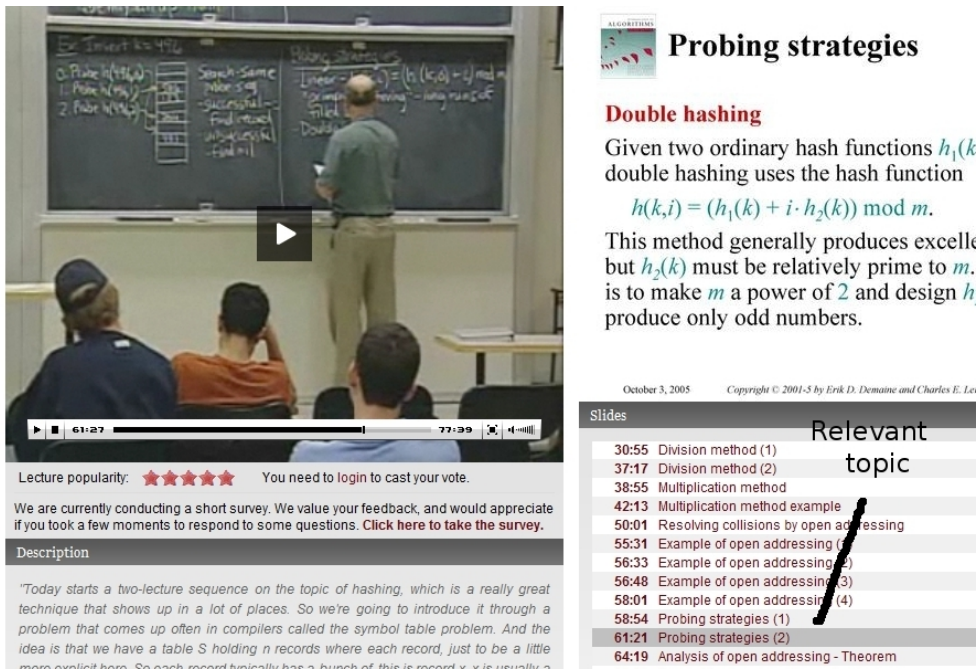
First video

- Duration - 61:22
- Found at - 42:32



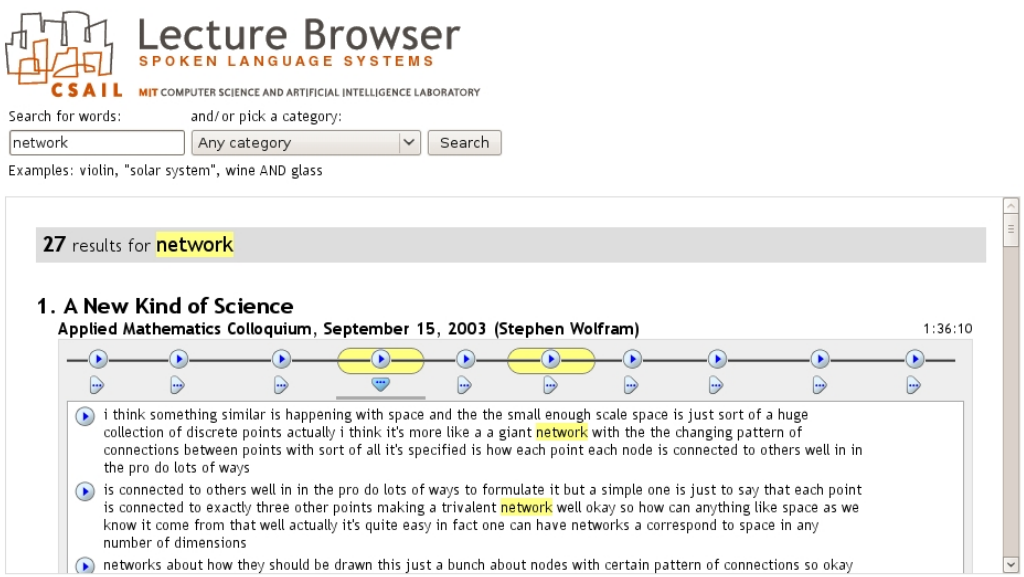
videolectures.net

- Provides free online access to lecture video recordings of various universities
- Has hyper links to slide change timings

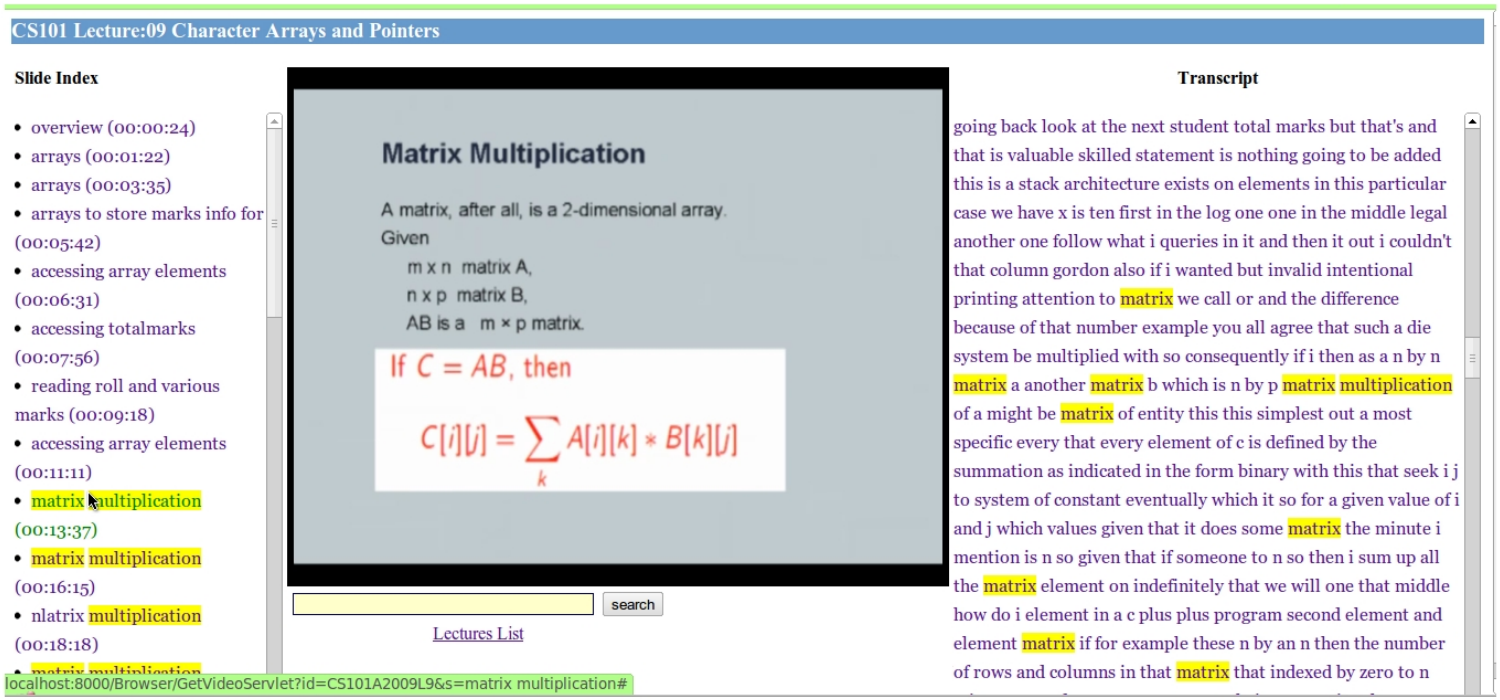


Lecture Browser

- Provides free on line access to lecture videos available in MIT Open Course ware
- Has Content based Search feature and highlights relevant segments of each video



Our System User Interface



Features in Lecture Video Repositories

Repository	Search	Navigation Features
CDEEP	No	No
NPTEL	No	No
freelecturevideos.com	Meta data	No
videolectures.net	Meta data	Slide Index (Manual)
Lecture Browser, MIT	Content	Speech Transcript
Our System	Content	Speech Transcript Slide Index (Automated)

Table: Lecture Video Repositories Comparison

Problems with existing systems

freevideolectures.com

- No indication of where exactly searched keywords occur within the video
- Takes more time to find required information

videolectuers.net

- Uses manual process for Synchronization of the slides

Why can't we use lecture browser?

- Can not be applied directly to our lecture videos.
- Requires speech recognition engine adaptation for non native english speakers
- Not an open source tool
- Their speech recognition engine is also not publicly available

How our system is different

- Provides automatic synchronization of slides.
- Improved user interface with more navigation features.
  It combines features in videolectures.net and lecture browser
- Open source application by integrating available speech recognition and text search engines
- Tune Sphinx speech recognition engine to recognize and transcribe Indian accents (English)

Outline

- Input: **keywords**
- Output :
  List of videos matching the keywords
  In each video portions where the keywords occur in the speech are highlighted
  When user clicks on a particular portion video starts playing in the media player
  Along with the media player user interface also shows slide index and speech transcript

- Scope of the project : Only deals with lecture videos which are in **English** and related **Computer Science** domain.
  Reason : Speech Recognition Engine

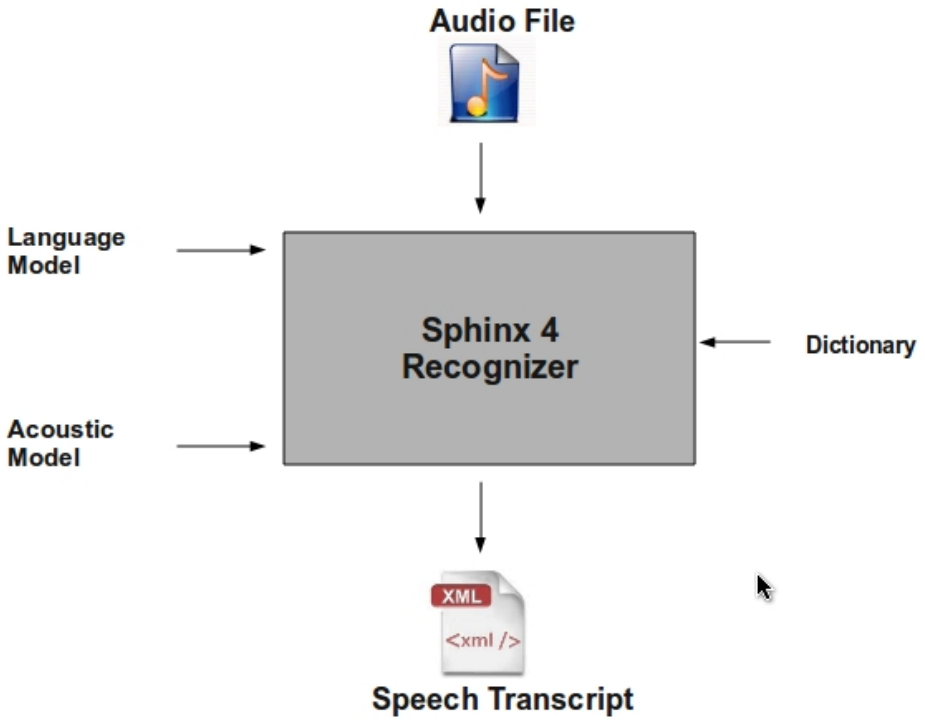


Figure: Sphinx 4 Recognizer

Steps in Speech Recognition

101011110 — Analog To Digital

101011101 → ST UH FF — Acoustic Modeling

Phonemes

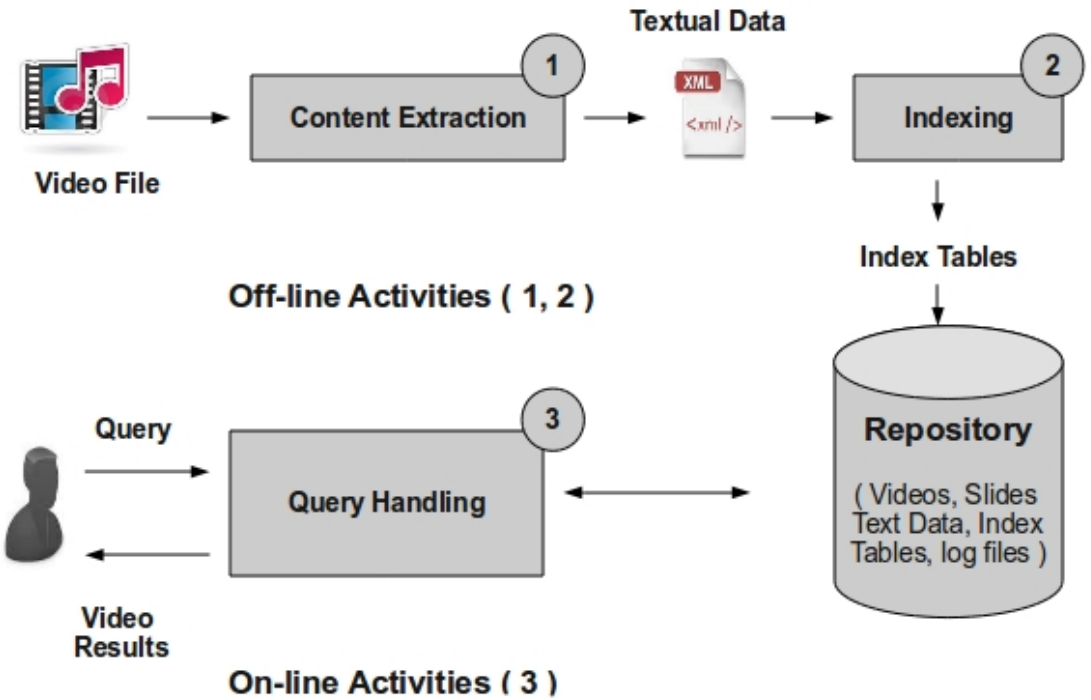
ST UH FF → stuff — Dictionary

This is book 0.89

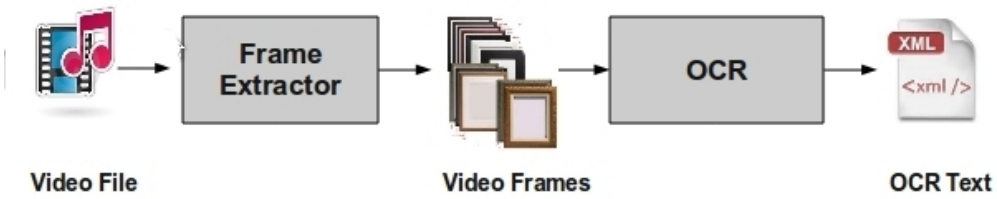
This it book 0.2 — Language Modeling

Outline

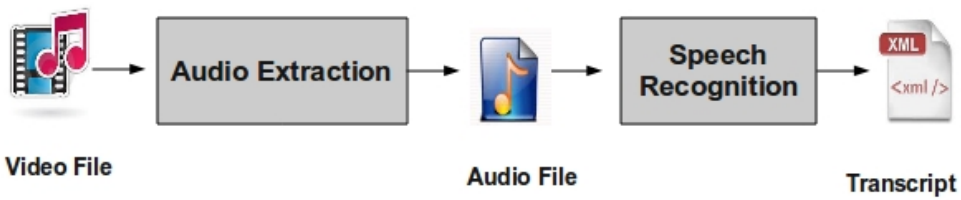
Solution Approach



Content Extraction



(a) Optical Character Recognition



(b) Speech Recognition

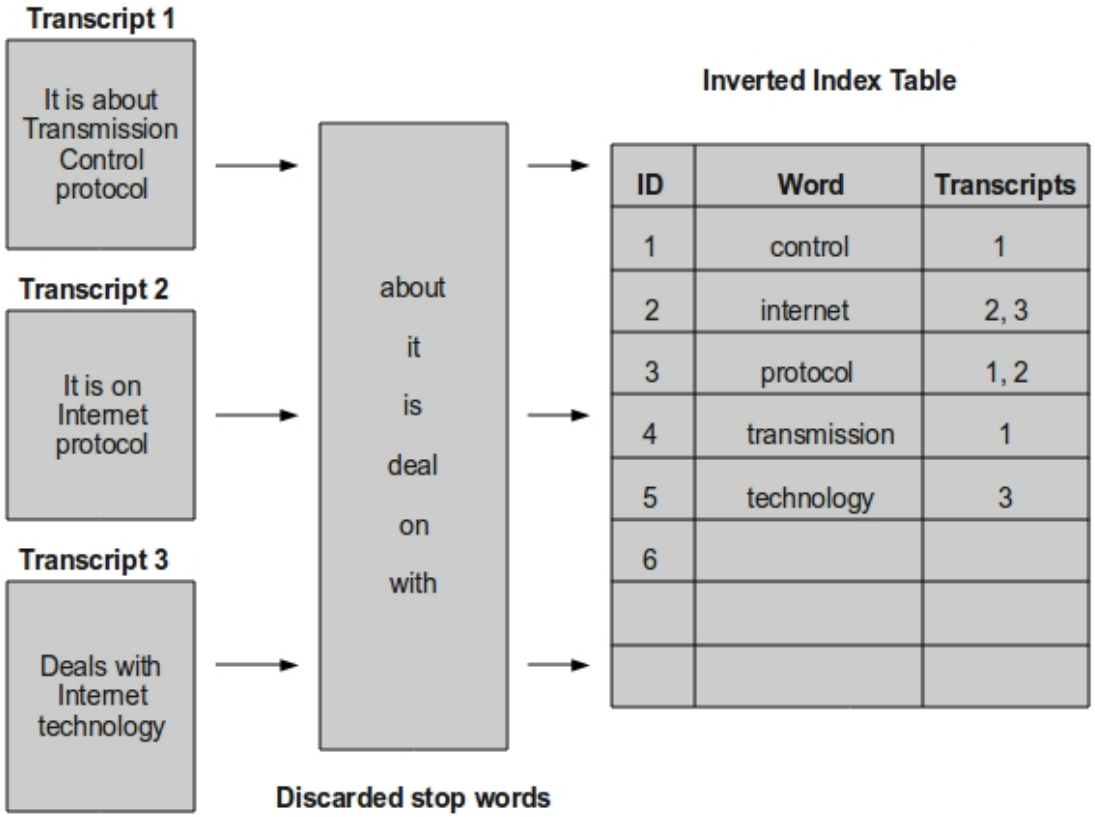
Speech Recognition Engines

- Sphinx 4 [18]
- Hmm Tool Kit (HTK) [9]

Reasons for choosing Sphinx

- Provides Java API(Application Programmable Interface)s, so it can be integrated easily into any application
- CMU Sphinx provides support for various tools useful in speech recognition
- Has easy configuration management where we need to set various parameters related to speech recognition
- Supporting tools are available for generation of acoustic and language models
- Completely written in java, it is highly modular and platform independent

Indexing & Query Handling



Text Search Engines

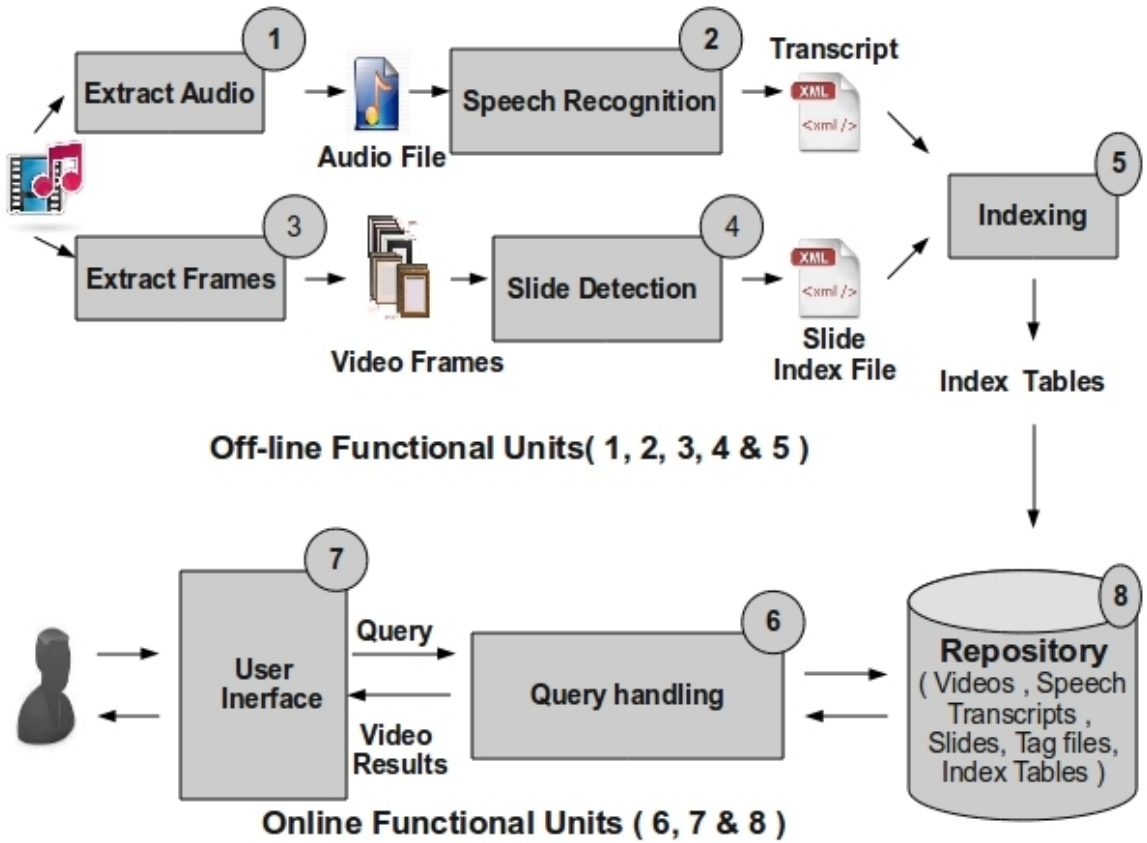
- Lucene[3], Indri[10]
- Xapian[22], Zettair[24]

Reasons for choosing Lucene

- It creates index of smaller size and search time is also very less[17]
- Supports ranked searching : best results returned first
- Can handle many powerful query types: phrase queries, wild card queries, range queries and more
- Mostly used text search engine. List of more than 150 applications and websites that are using Lucene to provide search facility[14]

Outline

System Components



Outline

Audio Extraction

- Input : Video file
- Output : Audio file
- Command line tools provided by FFmpeg [7]
- Running *ffmpeg* :
  $ *ffmpeg* -i *CS101_L10_Strings.mp4* -ar 16000 -ac 1 *CS101_L10_Strings.wav*

Speech Recognition

- Input : Audio file
- Output : Time aligned transcript in XML format
- Open source Java library for Sphinx-4 Speech Recognizer from CMU Sphinx [18]
- Requires language model, acoustic model and a pronunciation dictionary

Language model creation

- Large amount of text corpus related to the domain of speech recognition is required
- CMU SLM Toolkit [6] is useful for creating language model from the text corpus

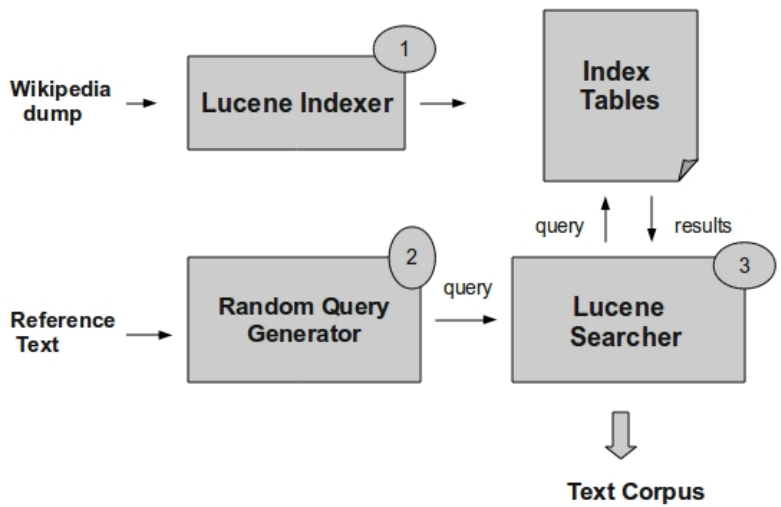


Figure: Framework for creating large amount of text corpus

Language model creation

- Collected text corpus related to Computer Science domain
- Wiki Index : Randomly generated queries consisting of terms from CS and searched in Lucene Indexes
- Text books : Data structures, Algorithms, Computer Networks, DBMS and OS
- Manual Transcriptions : Available in MIT OCW [4]
- Converted PDF files to Text using Java library provided from PDFBox [11]

Acoustic model development

- Requires audio files and corresponding manual transcriptions
- Developing new acoustic modeling takes large amount of time
- Adaptation of acoustic model is an option which requires an existing model
- CMU Sphinx provides WSJ and HUB4 models useful for recognizing US English
- Sphinx Train and Sphinx Base are set of tools useful for development for acoustic model

Acoustic model development

- We have to adapt an acoustic model to match our speakers to get better recognition accuracy
- Time consuming, which requires small audio files each having a sentence and manual transcription of each of the audio file
- Created 150 wav files for adaptation from CS101 lectures of Prof.Deepak Phatak
- Each of the wav file duration is 2 to 5 seconds and gave manual transcriptions for them

Speech Transcript Generation

- Configured the Sphinx-4 recognizer with the created language model and acoustic model
- Transcribed audio files of CS101 lectures and generated time aligned transcripts
- Transcribing of an audio file took approximately double the duration of the file
- The transcription speed can be increased but gives low recognition accuracy

Example Speech Transcript

```
<transcript>
  <tt>
     <text> deals with </text>
     <time> 7 </time>
  </tt>
  <tt>
    <text> searching </text>
    <time> 11 </time>
  </tt>
<tt>
     <text> of lectures </text>
     <time> 14 </time>
  </tt>
</transcript>
```

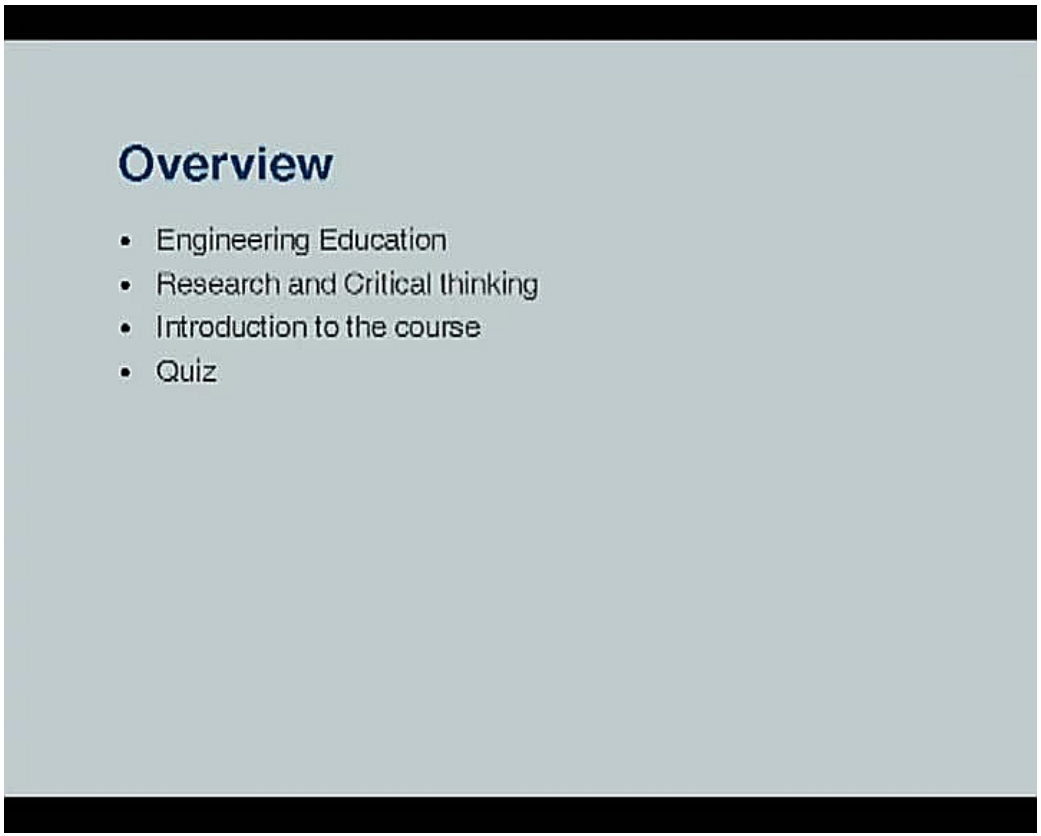
Video Frames Extraction

- Input : Video file
- Output : Frames extracted from the video at specified intervals
- *ffmpeg* can be used for the frame extraction
- $ *ffmpeg* -i *CS101_L10_Strings.mp4* -r 1 -f image2 *image_%4d.jpeg*

Slide Detection

- Input : Video frames of a lecture
- Output : Slides of the lectures along with their title and time of occurrences
- Designed an algorithm based on slide title matching which uses OCR for slide text extraction
- Found an OCR tool called tesseract-ocr [19] which gives better recognition accuracy among available the Open Source tools

Example frame from a video lecture



After applying OCR

```
Overview
 Engineering Education
 He$earchar1&iUrilmu| lhinkirng
 lnirucluctivn tc the course
 Oui;
```

Title Matching algorithm for Slide Detection

Title Time

---

overview 0104 $\longrightarrow$ Will be identified as starting of a slide
overview 0105
overview 0106
overview 0107
overview 0108
overview 0109
overview 0110
engineering 0135 $\longrightarrow$ Will be identified as starting of next slide
engineering 0136
engineering 0137
engineering 0138
engineering 0139
engineering 0140

Title Matching algorithm for Slide detection

```
while i < titles.length-1
begin
    if !titles[i].equals(prev) && matchesNextTwo(titles,i)
      indices.add(i);
      i = findNextSlide(titles,title[i],i+3)
  if i == -1
      return;
  endif
      prev = titles[i];
      indices.add(i);
      i = i + 2;
    endif
    i = i + 1;
end
```

Example Slide Index

```
<slides>
   <slide>
    <title> Overview </title>
    <time> 13 </time>
  </slide>
  <slide>
    <title> Introduction </title>
    <time> 79 </time>
  </slide>
</slides>
```

Indexing

- Input : Transcript file and Slide index file
- Output : Creates an Index or adds to existing indexes
- Apache Lucene [3] provides Java library for indexing text documents
- Parsed the transcript and slide index file which are in XML format
- Indexed CS101 lectures of Autumn 2009 and created indexes are of size 2.5MB

Query Handling

- Input : User given queries
- Output : List of lectures matching the query
- Apache Lucene [3] is also include Java classes for searching the indexes
- Technologies : Java Server Pages (JSPs) and Java Servlets
- Web Server : Apache Tomcat/6.0.24 [2]
- Operating System : Ubuntu Lucid Lynx 10.04 LTS

User Interface

- Created web pages using HTML and Java Script
- Using a freely available version of JW Player [12] for playing videos in the interface

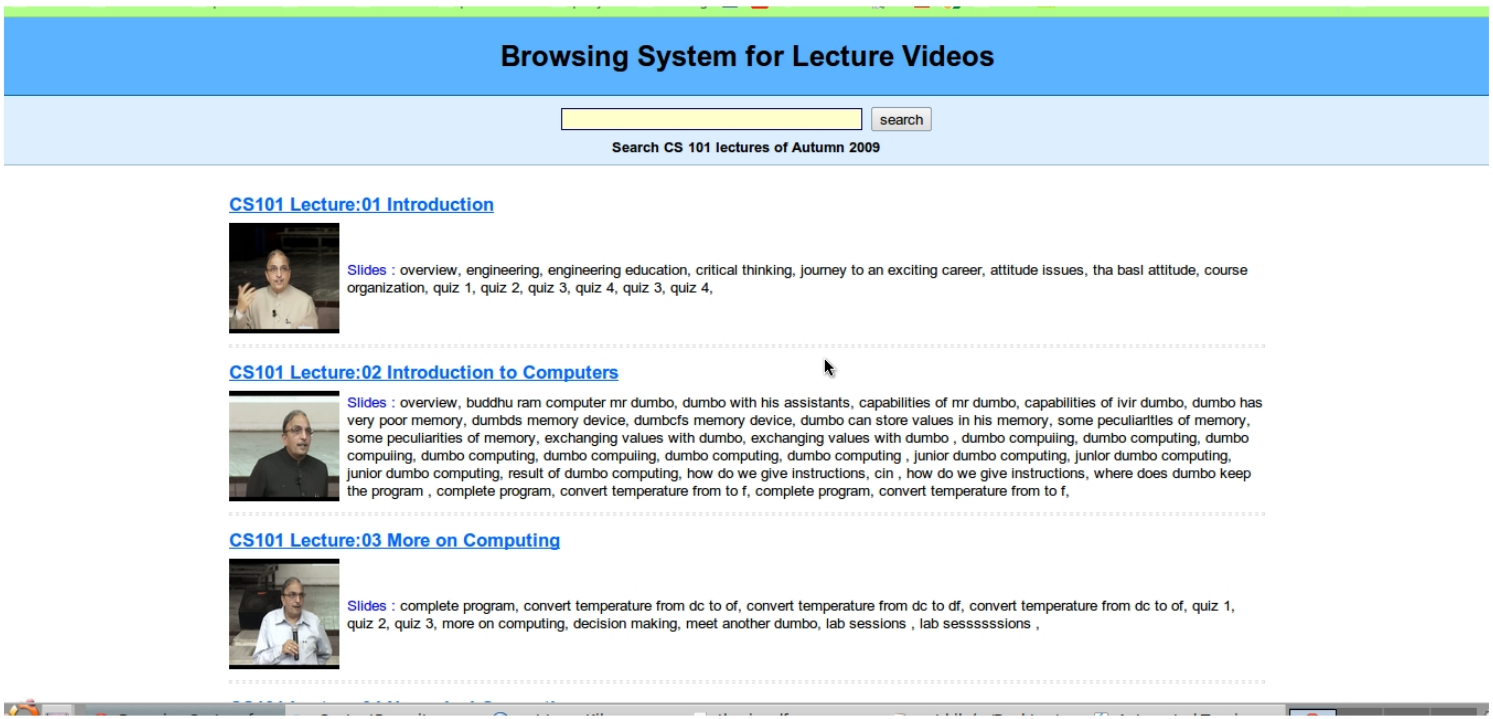


Figure: User Interface of our System

User Interface

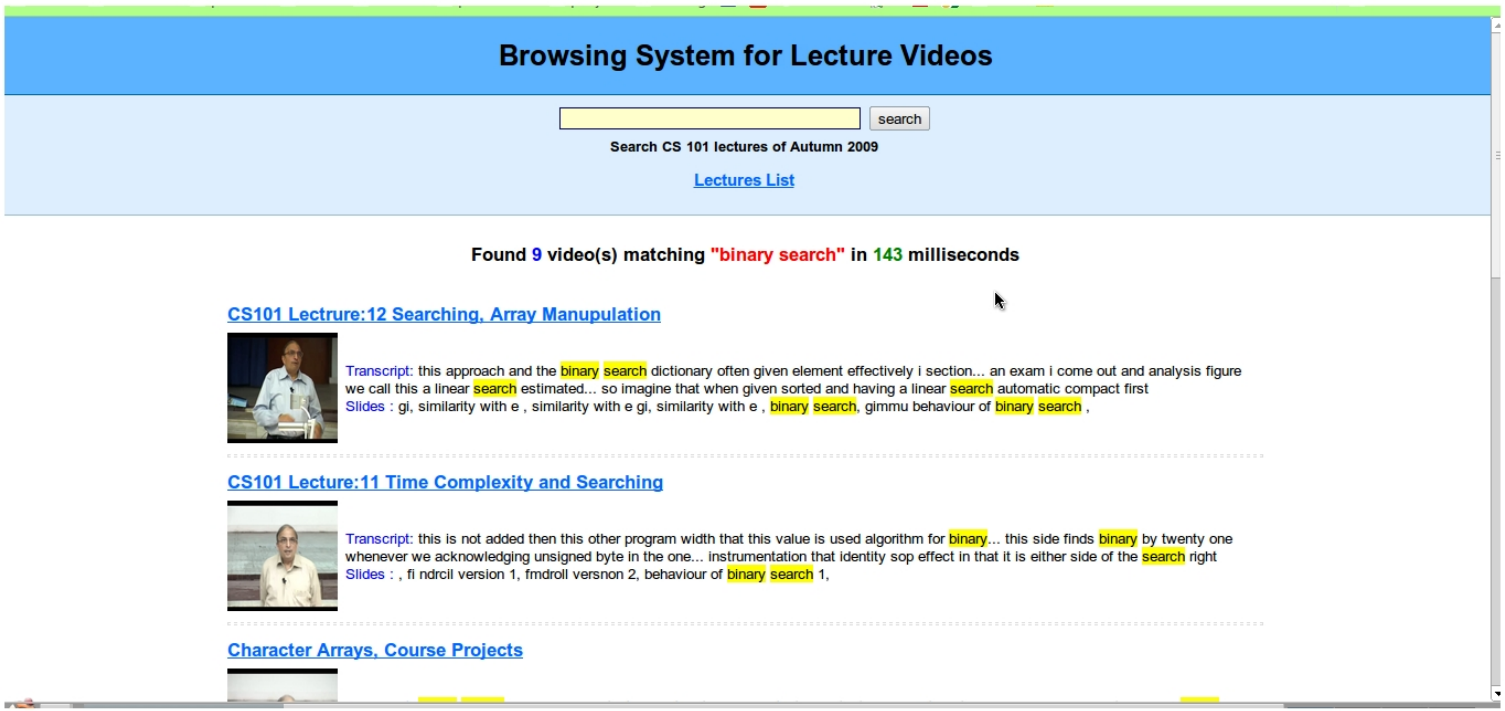


Figure: Search Results for query **binary search**

User Interface

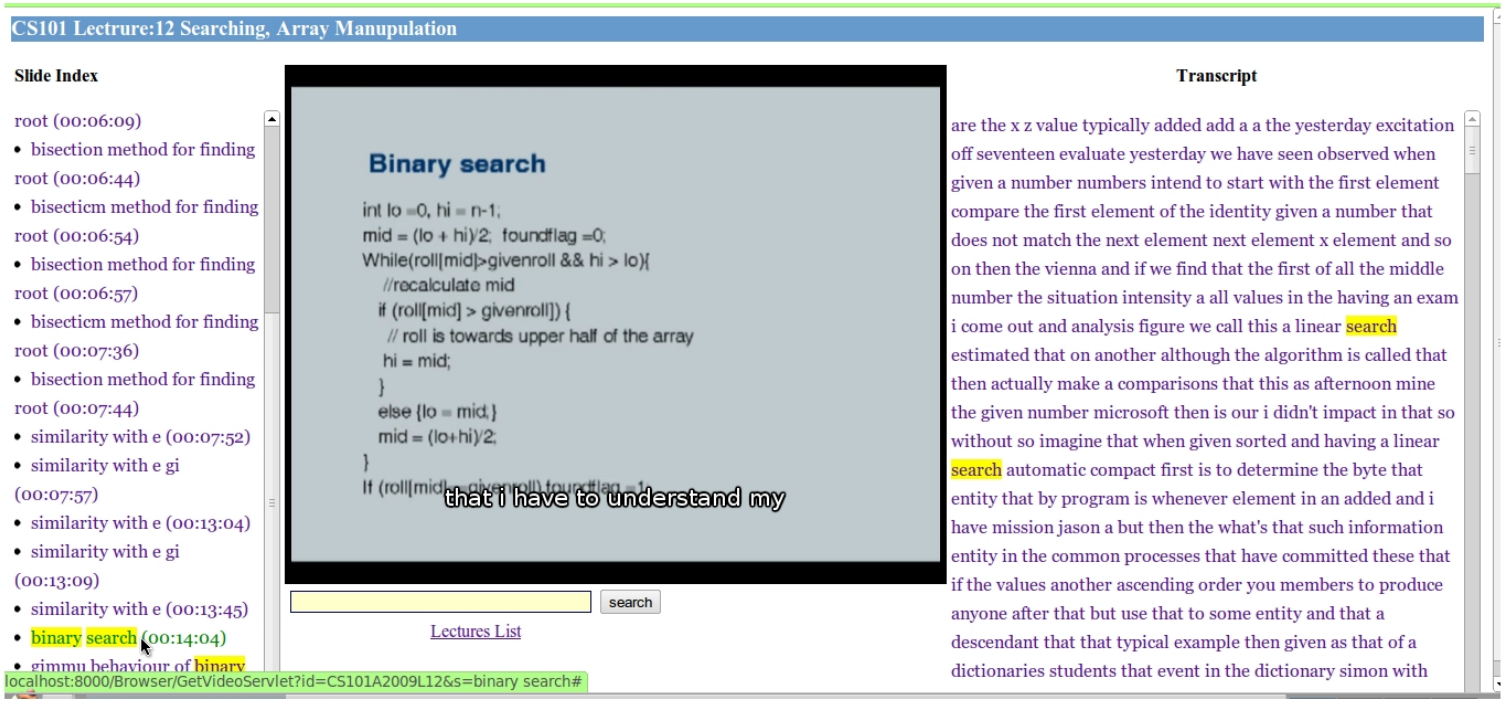


Figure: playing selected video with the navigation

Content Repository

- Recorded videos of lectures
- Speech transcripts
- Slide Index files
- Lucene indices

Outline

Slide Detection Results

Video	Actual slides	Detected slides	Correctly detected	Duplicates	Recall (%)	Prec. (%)
L_01	14	14	12	0	100	85
L_02	20	20	16	6	100	80
L_03	12	11	11	2	91.6	100
L_04	32	30	26	9	93.7	86.6
L_05	32	30	28	5	93.6	93.3
Total	110	105	93	18	95.4	88.5

Table: Slide Detection results

Speech Recognition Results

Adaptation files	Words in test files	Matches	Accuracy(%)
0	127	22	13
30	119	43	31
60	124	70	52
90	120	76	59
120	110	69	61
150	123	82	62

Table: Speech Recognition results

Video Retrieval Results

No.of queries tested	30
Avg Search seconds	0.004
Recall	0.72
Avg Precision	0.91

Table: Search Quality Results

Outline

Conclusion and Future Work

- Built a system for providing search facility in CS101 Autumn 2009 lectures
- Speech recognition accuracy can be improved through more adaptation
- Slide Detection method can be improved to reduce duplicate slides
- More lectures can be added to the repository


Academic Earth.
http://academicearth.org/.


Apache : An Open Source Web Server.
http://tomcat.apache.org/.


Apache Lucene.
http://lucene.apache.org/java/docs/index.html.


Audio/Video Lectures from MIT OCW.
http://ocw.mit.edu/courses/audio-video-courses/#electrical-engineering-and-computer-science.


CDEEP , IIT Bombay.
http://www.cdeep.iitb.ac.in/.


CMU Statistical Language Modeling Toolkit Documentation.
http://www.speech.cs.cmu.edu/SLM/toolkit_documentation.html/.


FFmpeg.
http://www.ffmpeg.org/.


freevideolectures.com.
http://www.freevideolectures.com/.


HTK.
http://htk.eng.cam.ac.uk/.


Indri.
http://www.lemurproject.org/indri/.


Java PDF Library.
http://pdfbox.apache.org/.


JW Player.
http://www.longtailvideo.com/players/jw-flv-player/.


Lecture Browser , MIT.

http://web.sls.csail.mit.edu/lectures/.


List of Applications that are using Lucene.
http://wiki.apache.org/lucene-java/PoweredBy.


List of educational video websites.
http://en.wikipedia.org/wiki/List_of_educational_video_websites.


nptel.
http://www.nptel.iitm.ac.in/.


Open Source Text Search Engines Evalution Results.
http://wrg.upf.edu/WRG/dctos/Middleton-Baeza.pdf.


sphinx.
http://www.speech.cs.cmu.edu/.


tesseract-ocr.
http://code.google.com/p/tesseract-ocr/.


videolectures.net.
http://www.videolectures.net/.


VideoPulp: Official Partners for Slide Index feature in NPTEL.
http://www.videopulp.in/.


xapian.
http://xapian.org/.


Youtube Edu.
http://www.youtube.com/education?b=400.


zettair.
http://www.seg.rmit.edu.au/zettair/.

Thank You