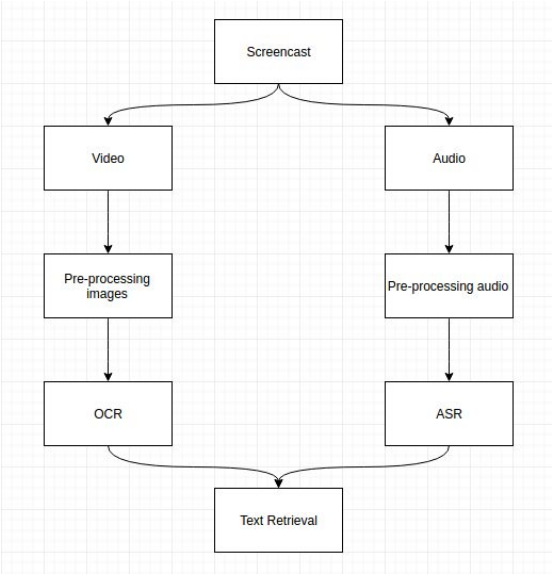
CA4009
“Screencast search”

Michael Wall - 13522003
Jordan Healy - 13379226

Introduction

- Video and audio search of lecture screencast
- User can search slides content and lectures speech
- Users can search by either audio, video or both
- Data collection taken from university libraries
- Cover all aspects of system;
  - OCR and its algorithm
  - ASR and its algorithm
  - Text Retrieval
  - Systems evaluation



Problem and motivation

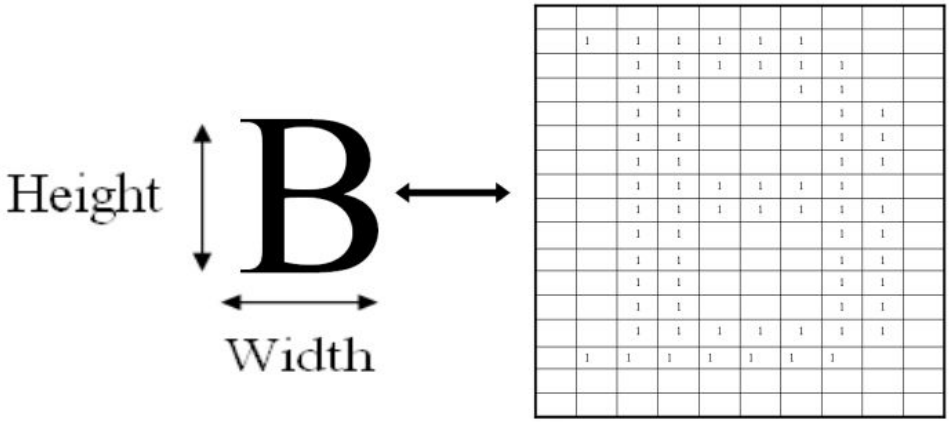
Information retrieval in long lecture presentation videos

- Time to find relevant information
- Knowledge of video relevancy
- Attention span of viewer

OCR - Optical character recognition

- Pre-processing: extract text from video frames to make them suitable for OCR
- Two algorithms for OCR
  - Matrix matching - comparing what the OCR sees with a library of character matrices
    - Pros: Easy to implement
    - Cons: Sensitive to noise, unaligned text and different styles of texts
  - Feature recognition - uses machine learning to looks for general features like open areas, closed shapes, diagonal lines and line intersections. Used for written text
    - Pros: More versatile
    - Cons: Harder to implement
- Chose matrix matching as it is typically used for typed text
  - Can assume presentation slides contain typed text
  - Feel we can overcome its cons in the pre-processing steps

OCR - Matrix Matching



- Have a database full of characters
  - (Character name, height, width, checksum, font)
  - “B”, 14, 8, 71, “Ariel”
- Calculate the height, width, and checksum for text in OCR
- Use strict matching to find exact values in database
- If not found, use soft matching to match values ±2 from database
- If character is found on an exact match, we can say with high certainty that the succeeding characters of that same font

Automatic Speech Recognition

YouTube API v3.0

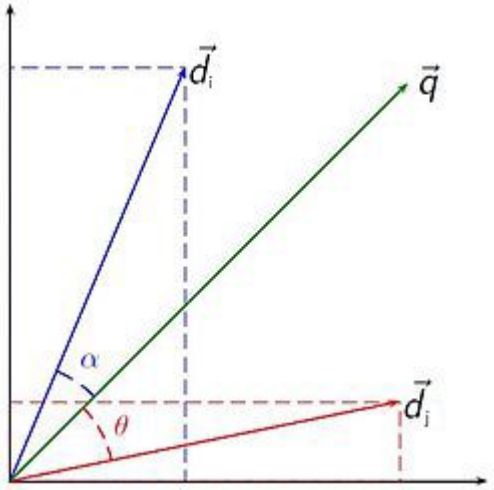
Captions:

- User uploaded
- Viewer submitted
- Google Speech-to-Text Engine

Text Retrieval

- Stop word removal - remove words like *a, of, the, and* from indexing
  - To improve search efficiency by not matching user query by stop words
- Stemming - remove suffixes of words in indexing
  - To improve efficiency by matching user query with different tenses of words
- Calculate term frequency - number of times a word occurs in a document
  - In long documents terms tend to be repeated which will skew the data
  - Use Document Length Normalisation to fix this

Vector Space Model



- Type of best-match search

Best-match	Exact-match
calculates a ranked list of a documents based on query	documents are either matches or non-matches

- Better than exact-match search for this reason
- Documents and queries are represented as vectors
- Similarity is the angle between vectors (calculated with cosine)
- Returns a ranked documents by relevance value sorted in descending order

Evaluation

Efficiency and effectiveness

- Maximizing MAP value
- Known search queries
- A/B testing w/ user feedback
- Time savings 40%

Any Questions?