CMU-SMU@TRECVID 2015: Video Hyperlinking

Zhiyong Cheng[1], Xuanchong Li[2], Jialie Shen[1], Alexander Hauptmann[2]

[1]Singapore Management University

[2]Carnegie Mellon University

Presented by Xuanchong Li

Outline

1. Introduction
2. Method
3. Experiment
4. Discussion

Motivation

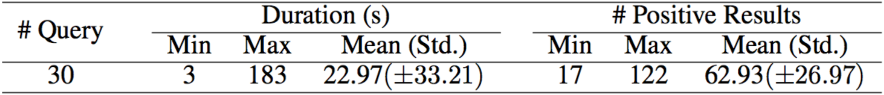
- Users are interested to find further information on some aspect of the topic of interest
- Link a video anchor or segment to other video segments in a video connection, based on similarity or relatedness
- We are first time to this task. Text-based methods are heavily used in previous work. We study more video-based methods/machine learning on this task.

Definition

Given a set of test videos with metadata with a defined set of anchors, each defined by start time and end time in the video, return for each anchor a ranked list of hyperlinking targets: video segments defined by a video ID and start time and end time. *– TRECVID 2015*

Dataset

- 2500-3500 hours of BBC video content
- Accompanied with metadata (title, short program descriptions and subtitles), automatic speech recognition (ASR) transcripts
- Training set: 30 query anchors with a set of ground-truth anchors are providedd



# Query	Duration (s)			# Positive Results		
	Min	Max	Mean (Std.)	Min	Max	Mean (Std.)
30	3	183	22.97(±33.21)	17	122	62.93(±26.97)

Methods Overview

- Mainly use text-based feature to get our best result
- Use text-bases feature with context information
- Use content-based feature (video, audio, etc.)
- Use various feature combination methods: linear weighted combination, learning to rank
- Categorize query into two groups

Pipeline

- Consider it as an ad-hoc retrieval problem
- Use fixed length (50s) video segmentation (It showed good performance in CUNI2014 video hyperlinking system)
- For each segment, different types of features are extracted and indexed
- For each extracted features, a variety of retrieval methods are explored
- Different strategies are used to combine the results obtained based on different features.
- Metrics: Precision@5, 10, 20, MAP, MAP bin, and MAP tol

Text-based Feature

- Subtitle
- ASR Transcription: LIMSI, LIUM, and NST-Sheffield
- Other metadata: title, short program descriptions and subtitles
- Context: 50s, 100s, 200s
- Combination of the above. e.g. 1. subtitle, 2. subtitle with 50s context, 3. subtitle with 100s context, 4. subtitle with 200s context, 5. subtitle and metadata, 6. subtitle and metadata with 50s context, 7. subtitle and metadata with 100s context and 8. subtitle and metadata with 200s context.

Retrieval Methods

- Use Terrier[2] IR system
- Use nine off-the-shelf methods: (1) BM25, (2) DFR version of BM25(DFR-BM25), (3) DLH hyper-geometric DFR model (DLH13), (4) DPH, (5) Hiemastras Language Model (Hiemastra-LM), (6) InL2, (7)TF-IDF, (8) LemurTF-IDF, and (9) PL2

Combining Text-based feature

- Weighted Linear Combination:

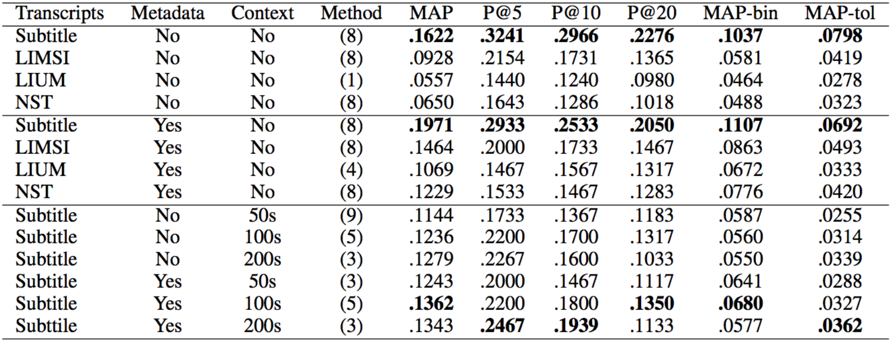
$$wlc(q, v) = w_1 \cdot rel(f_1) + w_2 \cdot rel(f_2) + \cdots + w_n \cdot rel(f_n) \qquad (1)$$

- Selected features are: Subtitle Metadata LemurTF-IDF, Subtitle Metadata DPH, Key Concept TF-IDF, improved trajectory and MFCC. Subtitle Metadata LemurTF-IDF
- Group the videos into two broad categories, train the weights separately:
  - Category 1: news & weather; science & nature; music (religion & ethics); travel; politics news; life stories music; sport (tennis); food & drink; motosport
  - Category 2: history; arts, culture & the media; comedy (sitcoms), cars & motors; antiques, homes & garden, pets & animals; health & wellbeing, beauty & style

Content-based Methods

- Feature:
  - Motion Feature: CMU Improved Dense Trajectory: 3 different versions.
  - MFCC: 2 different versions
  - Visual Semantic Feature from SIN task: 6 different versions
- Simply Taking linear distance as retrieval scores. Approximate linear space by explicit feature mapping.
- Learing to rank: retrain a model on the retrieval scores.

Experiment Results: Text-based Methods

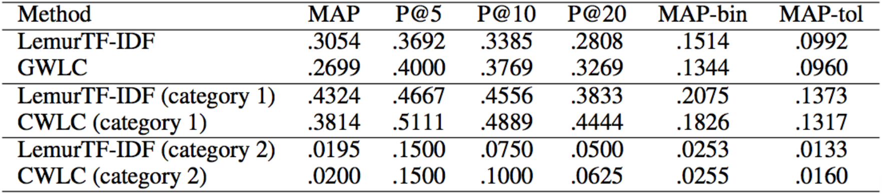


Transcripts	Metadata	Context	Method	MAP	P@5	P@10	P@20	MAP-bin	MAP-tol
Subtitle	No	No	(8)	**.1622**	**.3241**	**.2966**	**.2276**	**.1037**	**.0798**
LIMSI	No	No	(8)	.0928	.2154	.1731	.1365	.0581	.0419
LIUM	No	No	(1)	.0557	.1440	.1240	.0980	.0464	.0278
NST	No	No	(8)	.0650	.1643	.1286	.1018	.0488	.0323
Subtitle	Yes	No	(8)	**.1971**	**.2933**	**.2533**	**.2050**	**.1107**	**.0692**
LIMSI	Yes	No	(8)	.1464	.2000	.1733	.1467	.0863	.0493
LIUM	Yes	No	(4)	.1069	.1467	.1567	.1317	.0672	.0333
NST	Yes	No	(8)	.1229	.1533	.1467	.1283	.0776	.0420
Subtitle	No	50s	(9)	.1144	.1733	.1367	.1183	.0587	.0255
Subtitle	No	100s	(5)	.1236	.2200	.1700	.1317	.0560	.0314
Subtitle	No	200s	(3)	.1279	.2267	.1600	.1033	.0550	.0339
Subtitle	Yes	50s	(3)	.1243	.2000	.1467	.1117	.0641	.0288
Subtitle	Yes	100s	(5)	**.1362**	.2200	.1800	**.1350**	**.0680**	.0327
Subttile	Yes	200s	(3)	.1343	**.2467**	**.1939**	.1133	.0577	**.0362**

- Manual subtitle is better than ASR transcription
- Adding video metadata helps a little
- Using context information does not help

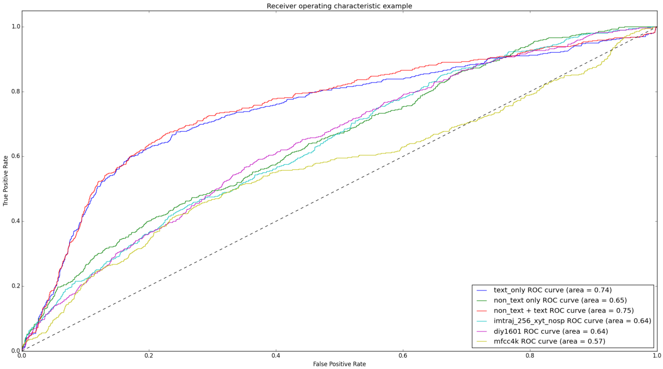
Experiment Results: Linear Combination of Text-based Feature

- Queries from Category 1 (more intra-class similarity) obtained much better results than queries from Category 2
- Performance decreases with the combination



Method	MAP	P@5	P@10	P@20	MAP-bin	MAP-tol
LemurTF-IDF	.3054	.3692	.3385	.2808	.1514	.0992
GWLC	.2699	.4000	.3769	.3269	.1344	.0960
LemurTF-IDF (category 1)	.4324	.4667	.4556	.3833	.2075	.1373
CWLC (category 1)	.3814	.5111	.4889	.4444	.1826	.1317
LemurTF-IDF (category 2)	.0195	.1500	.0750	.0500	.0253	.0133
CWLC (category 2)	.0200	.1500	.1000	.0625	.0255	.0160

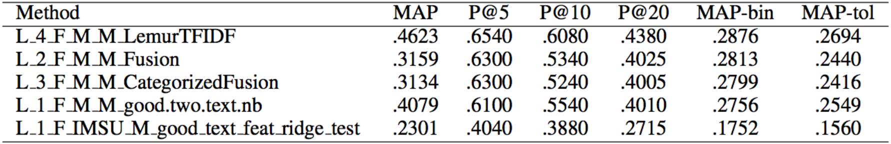
Experiment Results: Content-based Method



- Text-only ROC: 0.74 V.S. Text + non-text ROC: 0.75
- Works on development data. But badly on test data.
- Imbalanced data problem: positive/negative ratio in training is skewed to positive.

Submission

- Subtitle Metadata LemurTF-IDF
- Global Weighted Linearly Combination
- Categorized Weighted Linearly Combination
- Using learning to rank to fuse the best two text feature with Naive Bayes, where the prior is strongly biased to negative
- Using learning to rank to fuse the best two text feature with Ridge Regression



Method	MAP	P@5	P@10	P@20	MAP-bin	MAP-tol
L_4_F_M_M_LemurTFIDF	.4623	.6540	.6080	.4380	.2876	.2694
L_2_F_M_M_Fusion	.3159	.6300	.5340	.4025	.2813	.2440
L_3_F_M_M_CategorizedFusion	.3134	.6300	.5240	.4005	.2799	.2416
L_1_F_M_M_good.two.text.nb	.4079	.6100	.5540	.4010	.2756	.2549
L_1_F_IMSU_M_good_text_feat_ridge_test	.2301	.4040	.3880	.2715	.1752	.1560

Discussion

- Manual annotations (subtitle and metadata) $>$ ASR transcriptions $>$ video-content based features (audio, visual and motion features)
- Lacking of Labeled data makes machine learning difficult.
- How to handle imbalanced data?
- How to better combine feature? Learning to rank and weighted combining does not work well.
- Queries in different categories render very different performance. How to use this?
- How to definre similarity on different aspects?