

# Auto-Detecting Topic Transitions in SmartMOOCs

Brian Reinbold  
brianjr3@illinois.edu



# Adding Intelligence to MOOCs

- Massive Open Online Courses are revolutionizing education
- SmartMOOCs is a project to incorporate AI/ML into MOOCs to improve learner experience
- Lectures often contain subtopics that can be leveraged to enhance user experience
- SmartMOOCs currently assumes a topic transition every minute

## TOPICS

00:00:07

small techniqu larg web data  
use document search can index

00:01:04

well valu featur map can defin  
relev vector line function

00:02:00

well valu featur map can defin  
relev vector line function

00:03:00

decod got togeth key reduc  
word function count id  
document

00:04:01

decod got togeth key reduc  
word function count id  
document

00:05:02

decod got togeth key reduc  
word function count id  
document

# Methodology

- Follow Wang and Goutte 2018 to detect topic change by evaluating the similarity between two consecutive documents.
- Divide a lecture transcript into  $T$  documents where  $T = \frac{\text{Total Lecture Time}}{\text{Time Interval}}$
- Calculate cosine similarity of documents:
  - $D_{cos}(t) = 1 - \cos(TF(t), TF(t-1))$
  - $\cos(TF(t), TF(t-1)) = \frac{TF(t) \cdot TF(t-1)}{\|TF(t)\| \|TF(t-1)\|}$
- Detect breakpoints in the time series to detect topic transitions





Set Up



# Installing Environment and Project

- Install using conda

```
conda env create --name tis-project --file=environment.yml  
conda activate tis-project
```

- Or try installing from requirements.txt

```
pip3 install -r requirements.txt
```

- Then install project as a package

```
pip install -e .
```



# Running project

- Project was developed using [Visual Studio Code](#) and [Window Subsystem for Linux](#) with Ubuntu 20.04
- For a demo of key analysis, run the [Jupyter](#) notebook *notebooks/demo.ipynb*
  - Analysis over one lecture “Lesson 4.1: Probabilistic Retrieval Model: Basic Idea”
  - Contains sections to process raw transcript, build vocabulary/corpus set of words in transcript, estimate and evaluate breakpoints
- Could also run “notebooks/demo.py” if issues with Jupyter
- Makefile is used to keep track of project dependencies and more easily replicate entire project
  - make run





# Running Project with Docker

- Build image
  - `docker build -t tis .`
- Running container and running demo.py file
  - `docker run tis`
- Attaching terminal to container
  - `docker run -it tis sh`
  - `make run`



# Data Processing





# Processing Transcripts

- Transcripts is broken into several second intervals
- Need to avoid ending segments in the middle of a sentence when combining

## Raw Transcript File

1  
00:00:00,086 --> 00:00:07,516  
[SOUND]  
This

2  
00:00:07,516 --> 00:00:10,282  
lecture is about  
the Probabilistic Retrieval Model.

3  
00:00:10,282 --> 00:00:11,805  
In this lecture,

4  
00:00:11,805 --> 00:00:17,806  
we're going to continue the discussion  
of the Text Retrieval Methods.



# Cleaning Text

- Remove punctuation
- Remove stop words (e.g. the, at, a)
- Combine common n-grams

## Common N-Grams in TIS Corpus

Word	Frequency
text mining	106
vector space model	81
text retrieval	81
search engines	43
machine learning	41
time series	40
natural language processing	32
information retrieval	29
web search	29
maximum likelihood estimate	26
opinion mining	16
naive bayes	16
unigram language model	11
inverse document frequency	8



# Stem Words Using Porter Stemmer

- Porter stemmer removes common inflectional endings off English words
- Reduced total vocabulary in TIS corpora by 40%
- Leads to less sparse vectors

## Results of Porter Stemmer

word	stem
probability	probabl
probabilistic	probabilist
vector	vector
vectors	vector
word	word
words	word
computer	comput
computation	comput
computational	comput





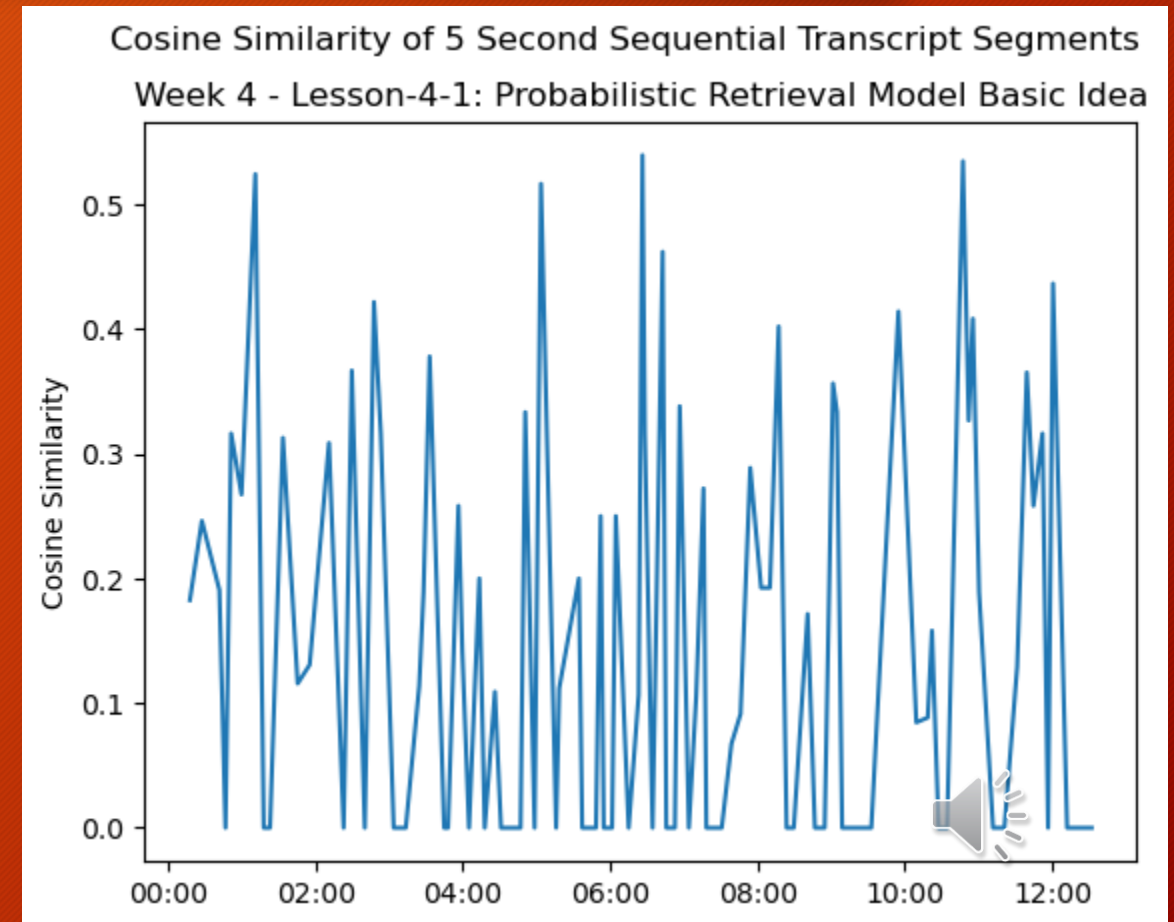
# Divide Transcript into Time Intervals

- Divide a lecture transcript into  $T$  documents where  $T = \frac{\text{Total Lecture Time}}{\text{Time Interval}}$
- Calculate cosine similarity of documents based on term frequency:
  - $D_{cos}(t) = 1 - \cos(TF(t), TF(t-1))$
  - $\cos(TF(t), TF(t-1)) = \frac{TF(t) \cdot TF(t-1)}{\|TF(t)\| \|TF(t-1)\|}$
- Need to combine sequential segments at large time intervals to reduce noise



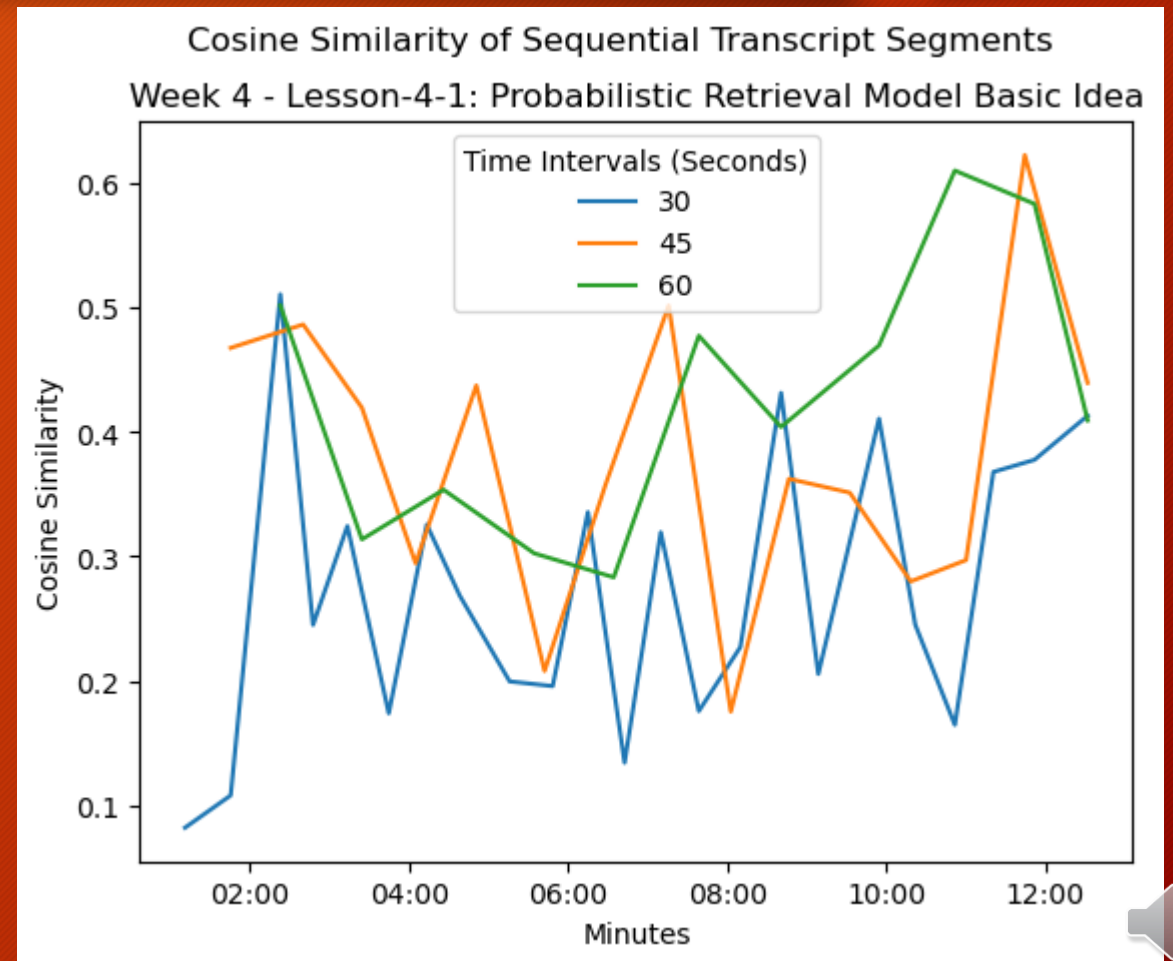
# Cosine Similarity of Sequential Segments at 5-Second Intervals

- Each segments contains nearly all unique token resulting in noise



# Cosine Similarity of Sequential Segments

- Tried 30, 45, and 60-second intervals
- Longer time intervals reduce noise in time series





# Detecting Topic Transitions

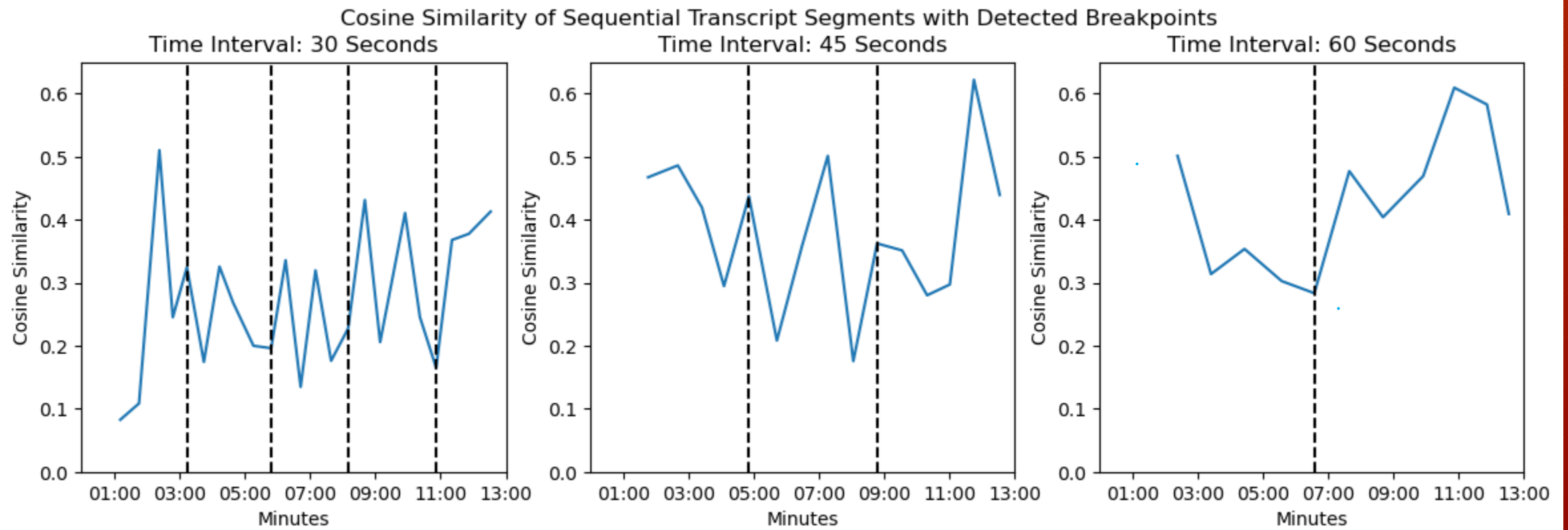


# Breakpoint Algorithm

- Used [ruptures](#) library for breakpoint algorithms
- Used linearly penalized segmentation based off [Killick 2012](#)
- Don't have to make any assumptions on how many breakpoints exist in the series



# Estimated Breakpoints





# Evaluation



# Silhouette Scores

- Metric minimizes intra-cluster distance and maximizes inter-cluster distances
  - Points within a cluster should be close together since they represent similar objects
  - Points in different clusters should be far apart since they represent distinct objects
- Value  $[-1, 1]$ 
  - 1: best value - well defined clusters
  - -1: worst value - points assigned wrong cluster
  - 0: implies overlapping clusters



# Silhouette Score Results

- Silhouette scores are mediocre
- Problem is clustering documents that are temporally correlated
- Still see improvement in scores over baseline so likely some benefit
- Imply that less subtopics are better

Summary of Silhouette Scores

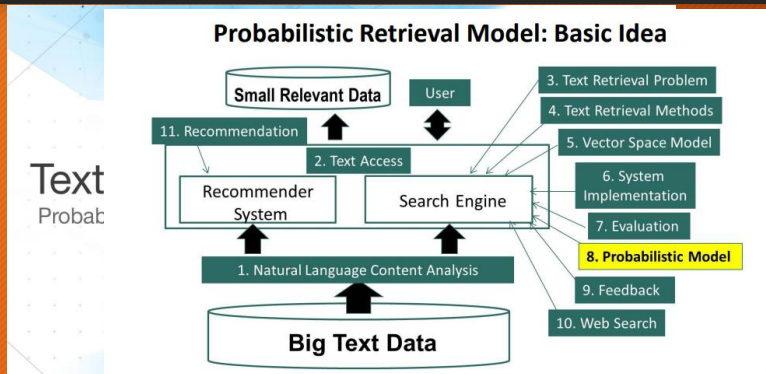
Time Interval	Average Number of Subtopics	Count of Best Scores	Average
Naïve, 60-Second Topic Transitions	13	2	-0.05149
30-Second Interval	4.7	5	-0.01627
45-Second Interval	3.2	27	-0.00179
60-Second Interval	2.4	62	0.006451





# Evaluate Specific Lecture:

## 4.1: Probabilistic Retrieval Model - Basic Idea



0:00 - 0:29

**Many Different Retrieval Models**

- **Probabilistic models:**  $f(d,q) = p(R=1 | d,q)$ ,  $R \in \{0,1\}$ 
  - Classic probabilistic model  $\rightarrow$  BM25
  - **Language model  $\rightarrow$  Query Likelihood**
  - Divergence-from-randomness model  $\rightarrow$  PL2

$$p(R=1 | d,q) \approx p(q | d, R=1)$$

If a user likes document  $d$ , how likely would the user enter query  $q$  (in order to retrieve  $d$ )?

0:29 - 2:54

**Probabilistic Retrieval Models: Basic Idea**

Query $q$	Doc $d$	Rel $R$
q1	d1	1
q1	d2	1
q1	d3	0
q1	d4	0
q1	d5	1
...	...	...
q1	d1	0
q1	d2	1
q1	d3	0
q2	d3	1
q3	d1	1
q4	d2	1

$f(q,d) = p(R=1 | d,q) = ?$

$$f(q,d) = p(R=1 | d,q) = ? \frac{\text{count}(q, d, R=1)}{\text{count}(q, d)}$$

$P(R=1 | q1, d1) = ? \frac{1}{2}$   
 $P(R=1 | q1, d2) = ? \frac{2}{2}$   
 $P(R=1 | q1, d3) = ? \frac{0}{2}$

What about unseen documents?  
Unseen queries?

2:54 - 8:32

**Query Likelihood Retrieval Model**

Query $q$	Doc $d$	Rel $R$
q1	d1	1
q1	d2	1
q1	d3	0
q1	d4	0
q1	d5	1
...	...	...
q1	d1	0
q1	d2	1
q1	d3	0
q2	d3	1
q3	d1	1
q4	d2	1

$f(q,d) = p(R=1 | d,q) \approx p(q | d, R=1)$

How likely the user enters  $q$

User likes  $d$

Assumption:  
A user formulates a query based on an "imaginary relevant document"

8:32 - 10:17

**Which doc is Most Likely the "Imaginary Relevant Doc"?**

$q = \text{"news about presidential campaign"}$

Documents and their likelihoods:

- d1:  $p(q|d1)$
- d2:  $p(q|d2)$
- d3:  $p(q|d4)$
- d4:  $p(q|d4)$
- d5:  $p(q|d5)$

The diagram shows the likelihood of the query  $q$  given each document  $d$  and its relevance  $R$ . The query is "news about presidential campaign". The documents are: d1 (news about ...), d2 (news about organic food campaign), d3 (news of presidential campaign), d4 (news of presidential campaign, presidential candidate), and d5 (news of organic food campaign, campaign, campaign).

10:17 - 10:59

**Summary**

- $\text{Relevance}(q,d) = p(R=1 | q,d) \rightarrow p(q | d, R=1)$
- **Query likelihood** ranking function:  $f(q,d) = p(q | d)$ 
  - Probability that a user who likes  $d$  would pose query  $q$
- How to compute  $p(q | d)$ ? How to compute probability of text in general?  $\rightarrow$  Language Model

$p(q = \text{"presidential campaign"} | d = \text{"... news of presidential campaign ... presidential candidate ..."})$

10:59 - 12:44

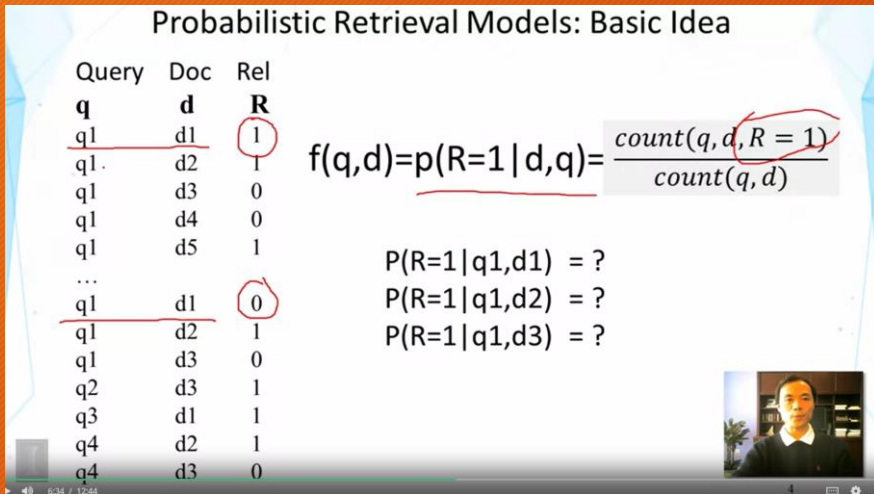
# Topic Transitions Based on Breakpoints from 60-Second Intervals

Probabilistic Retrieval Models: Basic Idea

Query	Doc	Rel
<b>q</b>	<b>d</b>	<b>R</b>
q1	d1	1
q1	d2	1
q1	d3	0
q1	d4	0
q1	d5	1
...		
q1	d1	0
q1	d2	1
q1	d3	0
q2	d3	1
q3	d1	1
q4	d2	1
q4	d3	0

$f(q,d)=p(R=1 | d,q)=\frac{\text{count}(q,d,R=1)}{\text{count}(q,d)}$

$P(R=1 | q1,d1) = ?$   
 $P(R=1 | q1,d2) = ?$   
 $P(R=1 | q1,d3) = ?$



6:34

Slide Title	End Time	Breakpoints of 60-Second Intervals
1-2: Intro/Outline	0:29	
3: Many Different Retrieval Models	2:54	
4: Probabilistic Retrieval Models: Basic Idea	8:32	6:34
5: Query Likelihood Retrieval Model	10:17	
6: Which doc is Most Likely the "Imaginary Relevant Doc"?	10:59	
7: Summary	12:44	





# Topic Transitions Based on Breakpoints from 45-Second Intervals

Probabilistic Retrieval Models: Basic Idea

Query	Doc	Rel
<b>q</b>	<b>d</b>	<b>R</b>
q1	d1	1
q1	d2	1
q1	d3	0
q1	d4	0
q1	d5	1
...		
q1	d1	0
q1	d2	1
q1	d3	0
q2	d3	1
q3	d1	1
q4	d2	1
q4	d3	0

$f(q,d)=p(R=1 | d,q)=?$

4:50

Query Likelihood Retrieval Model

Query	Doc	Rel
<b>q</b>	<b>d</b>	<b>R</b>
q1	d1	1
q1	d2	1
q1	d3	0
q1	d4	0
q1	d5	1
...		
q1	d1	0
q1	d2	1
q1	d3	0
q2	d3	1
q3	d1	1
q4	d2	1
q4	d3	0

$f(q,d)=p(R=1 | d,q) \approx p(q | d, R=1)$

User likes d

8:46

Slide Title	End Time	Breakpoints of 45-Second Intervals
1-2: Intro/Outline	0:29	
3: Many Different Retrieval Models	2:54	
4: Probabilistic Retrieval Models: Basic Idea	8:32	4:50
5: Query Likelihood Retrieval Model	10:17	8:46
6: Which doc is Most Likely the "Imaginary Relevant Doc"?	10:59	
7: Summary	12:44	





# Topic Transitions Based on Breakpoints from 30-Second Intervals

Probabilistic Retrieval Models: Basic Idea

Query	Doc	Rel
q	d	R
q1	d1	1
q1	d2	1
q1	d3	0
q1	d4	0
q1	d5	1
...		
q1	d1	0
q1	d2	1
q1	d3	0
q2	d3	1
q3	d1	1
q4	d2	1
q4	d3	0

$f(q,d)=p(R=1 | d,q)=?$

3:13

Probabilistic Retrieval Models: Basic Idea

Query	Doc	Rel
q	d	R
q1	d1	1
q1	d2	1
q1	d3	0
q1	d4	0
q1	d5	1
...		
q1	d1	0
q1	d2	1
q1	d3	0
q2	d3	1
q3	d1	1
q4	d2	1
q4	d3	0

$f(q,d)=p(R=1 | d,q)=\frac{\text{count}(q,d,R=1)}{\text{count}(q,d)}$

$P(R=1 | q1,d1) = ?$   
 $P(R=1 | q1,d2) = ?$   
 $P(R=1 | q1,d3) = ?$

5:48

Probabilistic Retrieval Models: Basic Idea

Query	Doc	Rel
q	d	R
q1	d1	1
q1	d2	1
q1	d3	0
q1	d4	0
q1	d5	1
...		
q1	d1	0
q1	d2	1
q1	d3	0
q2	d3	1
q3	d1	1
q4	d2	1
q4	d3	0

$f(q,d)=p(R=1 | d,q)=\frac{\text{count}(q,d,R=1)}{\text{count}(q,d)}$

$P(R=1 | q1,d1) = 1/2$   
 $P(R=1 | q1,d2) = 2/2$   
 $P(R=1 | q1,d3) = 0/2$

What about unseen documents?  
Unseen queries?

8:10

Which doc is Most Likely the "Imaginary Relevant Doc"?

q = "news about presidential campaign"

d1: ... news about ...  
d2: ... news about organic food campaign ...  
d3: ... news of presidential campaign ...  
d4: ... news of presidential campaign ...  
d5: ... news of organic food campaign ...

$p(q|d1)$ ,  $p(q|d2)$ ,  $p(q|d4)$ ,  $p(q|d5)$

10:51

Slide Title	End Time	Breakpoints of 30-Second Intervals
1-2: Intro/Outline	0:29	
3: Many Different Retrieval Models	2:54	3:13
4: Probabilistic Retrieval Models: Basic Idea	8:32	5:48 8:10
5: Query Likelihood Retrieval Model	10:17	
6: Which doc is Most Likely the "Imaginary Relevant Doc"?	10:59	10:51
7: Summary	12:44	



# Conclusion



# Does this Approach Work?

- Kind of
- Difficult to judge without human evaluation
  - Silhouette scores are not useful
- Some evidence of identifying subtopics
  - May not generalize and scale to all lectures though
- Breakpoints arbitrarily appear in middle of slides





# Next Steps

- Assume each slide is a subtopic and find times of each slide transition
  - Then repeat methodology to see if two slides are related
  - Downside is it may be sensible to divide slide into multiple subtopics
- Utilize word embeddings to reduce sparseness of term frequency vectors which should reduce noise



# References

- C. Truong, L. Oudre, N. Vayatis. Selective review of offline change point detection methods. *Signal Processing*, 167:107299, 2020.
- Killick, R., Fearnhead, P., & Eckley, I. (2012). Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500), 1590-1598.
- Wang, Yuli and Goutte, Cyril. “Real-time Change Point Detection using On-line Topic Models.” *Association for Computational Linguistics*, 2018, pp. 2505-2515.