11-642:
Search Engines

Search Log Analysis

Jamie Callan and Jon Elsas
Carnegie Mellon University
callan@cs.cmu.edu

1

Lecture Outline

- **Segmenting search logs into sessions**
- **Query suggestions**
- **Query intents**
- **Click models**

2

Information Seeking in the Real World

Interpreting search logs is an open research problem

- d_1 is clicked at steps 2 and 4 … is it relevant to q_1?
- Are q_1, q_2, and q_3 about the same information need?
- Was the user satisfied with any of the search results?

How do we think about this sequence of interactions?

Search log

$\boxed{q_1}$
d_1
d_2
d_1
$\boxed{q_2}$
$\boxed{q_3}$
d_3
email site
:

q_i: Query
d_j: Clicked page

Information Seeking is a Dialogue Between a Person and a Search Engine

Ad-hoc search can be viewed as a *dialogue* about an information need

Person: query	Initial description
Engine: search results	Initial attempt to satisfy it
Person: reformulated query	Revised description
Engine: new search results	Revised attempt to satisfy it
…	

Viewing Search Logs as a Dialogue

The first task is to distinguish the different dialogues
- Which queries address the same information need?

Originally, information need ≈ a search session
- **Session:** A sequence of user actions within a timespan
  – E.g., 30 minutes
- Perhaps an artifact of the experimental conditions
  – Much of the early work was done in a lab

Information need ≈ a search session is beginning to be challenged
- However, we start here because it is still the dominant view

Search log

$\boxed{q_1}$
d_1
d_2
d_1
$\boxed{q_2}$
$\boxed{q_3}$
d_3
email site
:

Viewing Search Logs as a Dialogue

gout	2006-03-01 07:38:03
chemotherapy side effects	2006-03-01 07:42:36
chemotherapy causing hearing loss	2006-03-01 07:45:23
kenny rogers songs	2006-03-02 06:05:40
commerce on line	2006-03-03 04:54:11
broadband internet	2006-03-06 05:32:28
middlesex county college nj	2006-03-06 16:55:56
kean college	2006-03-06 17:02:32
montclair college	2006-03-06 17:10:45
union county college	2006-03-07 04:49:23
rutgers	2006-03-07 05:10:17
kean college	2006-03-07 05:19:22
migraine headache	2006-03-10 06:02:55
new jersey income tax	2006-04-12 06:09:44

How would you segment this log into sessions?

(From AOL search log, part 9)

Segmenting Search Logs into Sessions: Simple Heuristics

Δ Time: Same session iff |timestamp (q_2) – timestamp| (q_1) < Δ

- Often Δ = 30 minutes, but many values have been tried
- Radlinski found 30 minutes to be effective in a library setting
- Jones found no value that is better than random on the web

Common term: Same session iff $q_1 \cap q_2 \neq \emptyset$

- Probably high Precision, low Recall

Rewrite classes: Common reformulation patterns

- E.g., term added, deleted, or replaced
- Probably high Precision, low Recall

Segmenting Search Logs into Sessions: Simple Heuristics

gout	2006-03-01 07:38:03	
		CT, RC
chemotherapy side effects	2006-03-01 07:42:36	
chemotherapy causing hearing loss	2006-03-01 07:45:23	
		ΔT, CT, RC
kenny rogers songs	2006-03-02 06:05:40	
		ΔT, CT, RC
commerce on line	2006-03-03 04:54:11	
		ΔT, CT, RC
broadband internet	2006-03-06 05:32:28	
		ΔT, CT, RC
middlesex county college nj	2006-03-06 16:55:56	
kean college	2006-03-06 17:02:32	
montclair college	2006-03-06 17:10:45	
		ΔT
union county college	2006-03-07 04:49:23	
		CT, RC
rutgers	2006-03-07 05:10:17	
		CT, RC
kean college	2006-03-07 05:19:22	
…		
		ΔT, CT, RC
migraine headache	2006-03-10 06:02:55	
		ΔT, CT, RC
new jersey income tax	2006-04-12 06:09:44a	

(From AOL search log, part 9)

Segmenting Search Logs into Sessions

Heuristics work surprisingly well

- **Task:** Do q_i and q_{i+1} describe the same information need?

Features	Accuracy
Predict 'same info need'	63.1%
30 minute threshold	57.2%
Trained time	69.5%
Common words	80.7%

(Jones and Klinker, 2006)

Segmenting Search Logs into Sessions: Other Features

gout	2006-03-01 07:38:03	
chemotherapy side effects	2006-03-01 07:42:36	CT, RC
chemotherapy causing hearing loss	2006-03-01 07:45:23	
		ΔT, CT, RC
kenny rogers songs	2006-03-02 06:05:40	
		ΔT, CT, RC
commerce on line	2006-03-03 04:54:11	

What other features could be used to segment a log?

- Δ time ≤ {5, 30, 60, 120} minutes
- Edit distance between queries
- Co-occurrence (e.g., PMI, χ^2) of queries in a query log
- Queries have co-occurring clicks in a query log
- ODP or Yahoo page category overlap of top 10 results
- Cosine or JSD similarity of top 10-50 results
- …

(Jones and Klinker, 2006)

Segmenting Search Logs into Sessions: Classifiers

A trained classifier is more effective than heuristics

Task: Do q_i and q_{i+1} describe the same information need?

Features	Accuracy
Levenshtein edit distance	85.0%
Time, common words	81.5%
Time, common words, cosine top 50 results	84.0%
All features	87.3%

Best trained time feature: Δ time $\leq$ 1.5 minutes

Metric: Classifier accuracy. Differences are statistically significant.

(Jones and Klinker, 2006)

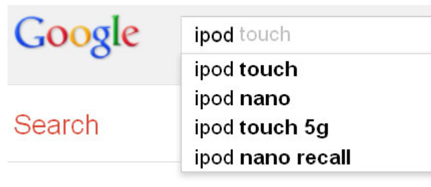
Segmenting and Organizing Query Logs

There is more recent work, but the main message hasn't changed

- **Predict whether two adjacent queries are for the same information need**
  - ~90% accuracy
- **Classifiers are best, but some heuristics aren't far behind**
  - Edit distance is very effective
  - Cosine distance among search results is effective
  - Time alone is primitive
    - » But effective in combination with other heuristics
    - » Still a very commonly-used heuristic

Lecture Outline

- Segmenting search logs into sessions
- **Query suggestions**
- Query intents
- Click models



Search Trails

A search trail is a single information seeking session

- One person, a series of queries, all within a short timespan
- Maybe the first query didn't find what the person wanted
- Maybe the sequence corresponds to query reformulations

Example	**Counterexample (?)**
ipod	ipod video discount
ipod nano	ipod video rebate
ipod nano recall	ipod video repair
ipod nano recall 2011	refurbished ipod video

Query Suggestions

Popular destinations for <u>frequent</u> queries can be looked up

… however, most queries are not frequent

- 57% of queries (20% of all searches) are unique

  97% of queries (66% of all searches) occur less than 10 times
  - *White, et al., 2007; 2008*

We cover two types of query suggestion approaches

- **Pseudo documents**
  - Works well for many (most?) queries
- **Co-occurrence statistics**
  - Works well for reasonably frequent queries

Query Suggestions #1: Pseudo Documents

Obtain <query$_i$, query$_{last}$> pairs from logs

- <“ipod”, “ipod nano recall 2011”>
- <“ipod nano”, “ipod nano recall 2011”>
- : : : :
- <“ipod repair”, “ipod nano recall 2011”>

Assume the last query in a session is successful

- Other success criteria is covered later in the course

Each query$_{last}$ is a candidate query suggestion

Query logs

: : : :

-- Session start --

ipod

ipod nano

ipod nano recall

ipod nano recall 2011

-- Session end –

: : : :

-- Session start --

ipod repair

ipod nano recall 2011

-- Session end –

: : : :

Query Suggestions #1: Pseudo Documents

Create a "pseudo document" for each query suggestion candidate

- The title is the query suggestion ($query_{last}$)
- The contents are the queries that preceded $query_{last}$ in a query log

Given a new query (e.g., 'ipod nano'), rank the suggestions

- Use your favorite retrieval model (e.g., BM25)

Pseudo documents are a general technique

- Used to rank all kinds of things besides queries

Pseudo document for a suggestion

<TITLE>
ipod nano recall 2011
</TITLE>
<BODY>
ipod, ipod nano,
ipod shuffle,
music players,
small ipod, ipod,
ipod micro, nanno,
buy ipod, …
</BODY>

17

17

Query Suggestions #2: Co-occurrence Statistics

Select queries that contain the user query q as a substring

- **Set_1:** 100 most frequent queries with q as a substring
- **Set_2:** 100 most frequent queries that followed q in a search session
- All queries in Set_1 and Set_2 are potential query suggestions q_s

$$Score(q_s) = \frac{Count(q_s) + \lambda_1}{N_1 + \lambda_1} \times \frac{Count_{follows}(q, q_s) + \lambda_2}{N_2 + \lambda_2}$$

q_s is frequent **q_s frequently follows q**

Jamie's understanding of N_1 and N_2 (the papers are vague)

- N_1: Sum of Count(q_s) for all candidates
- N_2: Sum of Count(q_s, q) for all candidates

(White, et al., 2007; 2008)

18

18

Query Suggestions #2: Co-occurrence Statistics

Select queries that contain the user query q as a substring

- **Set_1:** 100 most frequent queries with q as a substring
- **Set_2:** 100 most frequent queries that followed q in a search session
- All queries in Set_1 and Set_2 are potential query suggestions q_s

$$Score(q_s) = \frac{Count(q_s) + \lambda_1}{N_1 + \lambda_1} \times \frac{Count_{follows}(q, q_s) + \lambda_2}{N_2 + \lambda_2}$$

q_s is frequent **q_s frequently follows q**

This baseline method does as well as two experimental methods

(White, et al., 2007; 2008)

19

19

Query Suggestions: Summary

Three methods for generating 'related query' suggestions

- Pseudo documents
  - Almost any query
- Co-occurrence
  - Common queries

Searches related to march madness

wisconsin badgers vs coastal carolina chanticleers
kentucky wildcats vs west virginia mountaineers
wisconsin badgers vs oregon ducks
kentucky wildcats vs notre dame fighting irish
march madness **schedule**
march madness **predictions**
march madness **winners**
march madness **2013 locations**

20

20

Lecture Outline

- Segmenting search logs into sessions
- Query suggestions
- **Query intents**
- Click models

Many Queries Have Multiple Intents

We have discussed the different intents behind some queries

- **jaguar:** A car, an animal, an operating system, …
- **flash:** Software, a superhero, part of a camera, …
- **mercury**: An element, a planet, a god, a car, …
- **michael jordan:** An athlete, a professor, a businessman, …
- **ai:** Artificial intelligence, Americal Idol, art institute, …

Query suggestions are one possible source of query intents

- Can they be inferred in other ways?

(Radlinski, et al., 2010)

Identifying the Most Common Intent for a Query

For a query q

- Identify q's neighborhood **("expand step")**
  - Identify the 10 most common reformulations q' of q
  - Identify the 10 most common reformulations q'' of each q'
- Reduce the neighborhood to the most related queries **("filter")**
- **Cluster** the queries to find intent groups
- **Estimate the popularity** of each query and intent group
- The **name** of an intent group is its most popular query

(Radlinski, et al., 2010)

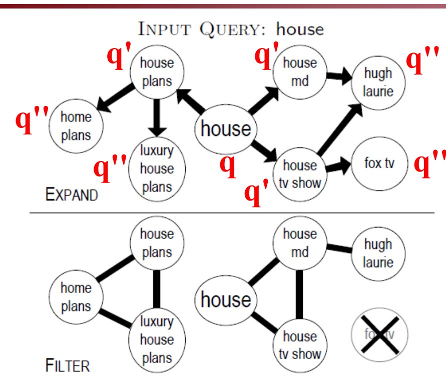
Identifying the Most Common Intent for a Query

Expand: Identify possibly-related queries

- (q, q') for at least 2 users
- $q'_{time} - q_{time} < 10$ min
- (q, q') must occur $\geq \delta$ times
  - Filters out frequent q' (e.g., gmail)

Filter: Improve precision

- Connect (q, q') if often clicked for same d
  - Removes many q'
  - May add some new q'
- Remove components with < 2 members



(Radlinski, et al., 2010)

Identifying the Most Common Intent for a Query

Cluster: Find groups with same intent

- E.g., use your favorite algorithm

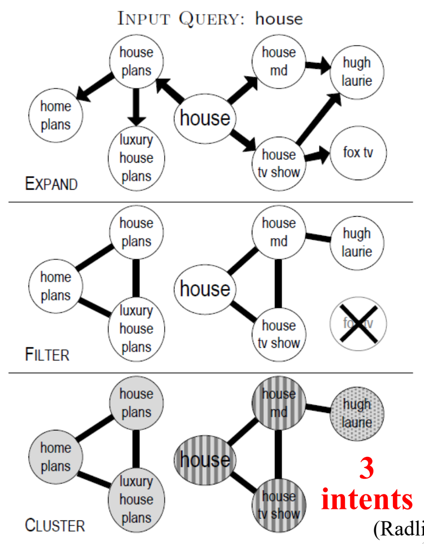
Estimate popularity

- Random walk for two iterations

$$w_q = 1$$

$$w_{q'} = \frac{w^q \cdot count(q \to q')}{\sum_{r \in R(q)} count(q \to r)}$$

- R(q): the set of all related queries q'
- Cluster score: Sum of query weights

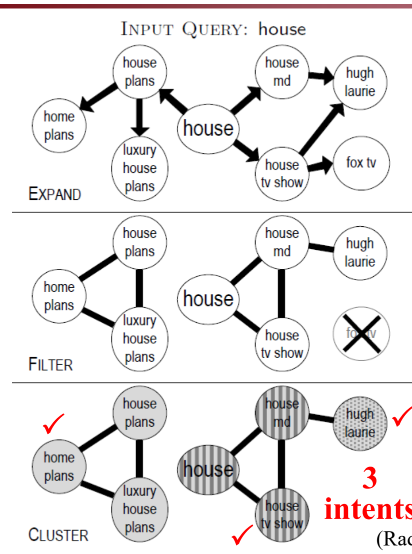


(Radlinski, et al., 2010)

Identifying the Most Common Intent for a Query

Name the intent group

- Use its highest-scoring query



(Radlinski, et al., 2010)

Identifying the Most Common Intents for a Query

juvenile delinquency

- juvenile delinquency ($w_c = 1.16$)
- causes of juvenile delinquency ($w_c = 0.50$)
- delinquency prevention ($w_c = 0.25$)
- definition of juvenile delinquency ($w_c = 0.20$)
- articles on juvenile delinquency ($w_c = 0.18$)
- reasons for juvenile delinquency ($w_c = 0.15$)

w_c: estimated relative popularity of each intent cluster

Intents created by TREC analysts

- What are the rates of juvenile crime in various jurisdictions, what is the nature of the offenses, how are they punished, and what measures are taken for prevention?

(Radlinski, et al., 2010)

27

27

Identifying the Most Common Intents for a Query

physical therapists

- physical therapist ($w_c = 1.22$)
- physical therapists salary ($w_c = 0.80$)
- how to become a physical therapist ($w_c = 0.21$)
- physical therapy schools in california ($w_c = 0.15$)
- physical therapist school of california ($w_c = 0.11$)
- physical therapist assistance programs ($w_c = 0.11$)

w_c: estimated relative popularity of each intent cluster

Intents created by TREC analysts

- How can I obtain information about training, licensing, and skills needed for physical therapists?

(Radlinski, et al., 2010)

28

28

Identifying the Most Common Intents for a Query

wireless communications

- wireless communications ($w_c = 1.07$)
- what is wireless comm. ($w_c = 0.56$)
- wireless comm. systems ($w_c = 0.19$)
- history wireless technology ($w_c = 0.13$)
- wireless cell phone companies ($w_c = 0.13$)
- wireless broadband providers ($w_c = 0.10$)

w_c: estimated relative popularity of each intent cluster

Intents created by TREC analysts

- Information on existing and planned uses, research/technology, regulations and legislative interest.

(Radlinski, et al., 2010)

Identifying the Most Common Intents for a Query

Key ideas

- Reformulation patterns + click information can be combined to identify common intents for ambiguous queries
- An ambiguous query may have several common intents
  - Not a surprise ☺
  - An intent is expressed by a group of queries with the same goal
- Popular intents may differ from what well-informed people expect

Lecture Outline

- Segmenting search logs into sessions
- Query suggestions
- Query intents
- **Click models**

Click Models

Typical search behavior

- A person issues a query q
- A **s**earch **e**ngine **r**esult **p**age (SERP) is returned
- The person examines the SERP
  - Maybe the person clicks on one or more links
- The person stops interacting with the SERP
  - Perhaps issues a new query
  - Perhaps moves on to a new task

(Chuklin, et al., 2016)

Click Models

Click models represent user search behavior as a <u>sequence of observed and hidden events</u>

- E: An item on the SERP is **<u>e</u>**xamined **Observed**
  - May depend on the document's rank (position)
- A: User is **<u>a</u>**ttracted by the item's representation **Hidden**
  - May depend on the snippet quality
  - May depend on the document's relevance to the query
- C: An item is **<u>c</u>**licked **Observed**
- S: The information need is **<u>s</u>**atisfied **Hidden**

(Chuklin, et al., 2016)

33

33

Click Models

Click models define dependencies among events

- E.g., p(E | rank=r): Probability of <u>e</u>xamining page at rank <u>r</u>
- E.g., p(C | E): Probability that an <u>e</u>xamined page is <u>c</u>licked

Learn model parameters from a search log

(Chuklin, et al., 2016)

34

34

Random Click Model (RCM)

Random Click Model

- Assume each document (url) u is equally likely to be clicked

  p $(C_u) = \rho$ Probability of clicking document (url) u

- Learn ρ from data

Maximum likelihood estimate of ρ

$$\rho = \frac{1}{\sum_{s \in S} |s|} \sum_{s \in S} \sum_{u \in s} c_u^s$$

S: Search sessions

c_u^s: Document (url) u was clicked in session s

The RCM is not very accurate

(Chuklin, et al., 2016)

Click-Through Rate (CTR) Models: RCTR

Rank-Based CTR (RCTR)

- The click likelihood depends upon the rank r

  p $(C_r) = \rho_r$ Probability of clicking document at rank r

- Learn ρ_r for each rank r

Maximum likelihood estimate of ρ_r

$$\rho_r = \frac{1}{|S|} \sum_{s \in S} c_r^s$$

S: Search sessions

c_r^s: Document at rank r was clicked in session s

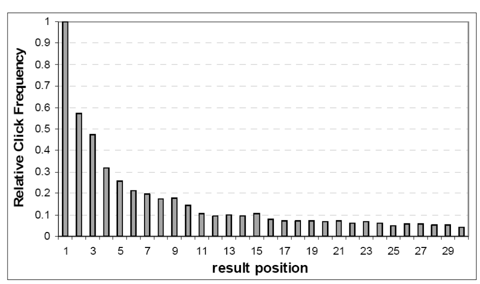


Figure 3.1: Relative click frequency for top 30 result positions over 3,500 queries and 120,000 searches.

(Agichtein, et al, 2006)
(Chuklin, et al., 2016)

Click-Through Rate (CTR) Models: DCTR

Document-Based CTR (DCTR)

- Estimate a click-through rate for every query-document pair

  $p(C_u) = \rho_{uq}$ Probability of clicking document u for q
- Learn ρ_{uq} for each query-document pair

Maximum likelihood estimate of $\boldsymbol{\rho_{uq}}$

$$\rho_{uq} = \frac{1}{|S_{uq}|} \sum_{s \in S_{uq}} c_u^s$$

S_{uq}: Sessions for query q that contain document u

c_u^s: Document u was clicked in session s

DCTR is prone to overfitting

- Little or no data for many document-query pairs

(Chuklin, et al., 2016)

37

37

Attractiveness

Some models incorporate the document attractiveness

- Usually attractiveness is a measure of snippet quality
- Usually $p(A_u)$ is independent of rank r

  $p(A_u) = \alpha_{uq}$ Probability that document u is considered attractive

(Chuklin, et al., 2016)

38

38

Position-Based Model (PBM)

Position-Based Model (PBM)

- Click probability depends on examination and attractiveness

  $p\,(C_u) = p\,(E_u) \cdot p\,(A_u)$

  $p\,(E_u) = \gamma_{r_u}$ Document at rank r_u is examined

  $p\,(A_u) = \alpha_{uq}$ Document u is attractive for query q

Model parameters can be estimated using EM

$$\alpha_{uq} \quad \alpha_{uq}^{(t+1)} = \frac{1}{|\mathcal{S}_{uq}|} \sum_{s \in \mathcal{S}_{uq}} \left(c_u^{(s)} + \left(1 - c_u^{(s)}\right) \frac{\left(1-\gamma_r^{(t)}\right)\alpha_{uq}^{(t)}}{1-\gamma_r^{(t)}\alpha_{uq}^{(t)}} \right),$$

$$\text{where } \mathcal{S}_{uq} = \{s_q : u \in s_q\}$$

$$\gamma_r \quad \gamma_r^{(t+1)} = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \left(c_u^{(s)} + \left(1 - c_u^{(s)}\right) \frac{\left(1-\alpha_{uq}^{(t)}\right)\gamma_r^{(t)}}{1-\gamma_r^{(t)}\alpha_{uq}^{(t)}} \right)$$

(Chuklin, et al., 2016)

39

39

Cascade Model (CM)

Assume that the user examines the SERP from top to bottom until they find a relevant document

- $p\,(C_r) \leftrightarrow p\,(E_r) \cdot p\,(A_r)$ Clicked ↔ Examined & Attractive
- $p\,(E_r) = 1$ First rank is always examined
- $p\,(A_r) = \alpha_{r_u q}$ Attractiveness depends on rank & query
- $p\,(E_r \mid \neg E_{r-1}) = 0$ Sequential examination
- $p\,(E_r \mid E_{r-1}, \neg C_{r-1}) = 1$ Keep examining until click
- $p\,(E_r \mid C_{r-1}) = 0$ Stop examining after a click

Can only describe sequential examination with a single click

- Less general that the position-based model (PBR)

(Chuklin, et al., 2016)

40

40

Cascade Model (CM)

Maximum likelihood estimate for CM

$\alpha_{uq} = \frac{1}{|S_{uq}|} \sum_{s \in S_{uq}} c_u^s$

S_{uq}: Sessions for query q that contain document u

Each session *s* is truncated at its first click

c_u^s: Document u was clicked in session *s*

41

41

Click Models

There are many click models ...

- E.g., consider time spent examining a SERP item (e.g., TACM)
  - Time between click and subsequent click
- E.g., consider scrolling behavior
- E.g., consider eye movement behavior
- …

None seem dominant at this point

- DCTR is popular because ρ_{uq} can be treated as an (overfitted) relevance score for training LeToR
- Time-Aware Click Model (TACM) claims to correlate well with human assessors

(Chuklin, et al., 2016)

42

42

Click Models

How accurate are different models?

- WSCD 2012 dataset (Yandex), 1 million sessions

Model	Log-likelihood	Perplexity	Cond. perplexity	Time (sec)
RCM	−0.3727	1.5325	1.5325	**2.37**
RCTR	−0.3017	1.3730	1.3730	2.45
DCTR	−0.3082	1.3713	1.3713	9.39
PBM	−0.2757	1.3323	1.3323	77.95
CM	$-\infty$	1.3675	$+\infty$	12.17

The Position-Based Model (PBM) is best of this group

- On this dataset, the best models are about 10% better than PBM

(Chuklin, et al., 2016)

43

43

Click Models

Click models have several uses

- Improve understanding of user behavior
  - E.g., model what SERP characteristics affect user behavior
- Guide development of better evaluation metrics
  - E.g., something more realistic than NDCG
- Measure deviation of observed behavior from 'typical' behavior
  - E.g., people click on this document much more than expected
- Generate realistic artificial data
  - E.g., for testing software
  - E.g., infer relevance from clicks
- Very important in web-based advertising

(Chuklin, et al., 2016)

44

44

Click Models

Click models can be used to generate artificial clicks

Input: Click model M, document ranking s

Output: Vector of simulated clicks $c_{1 \ldots} c_r$

Algorithm:

for rank = 1 to $|s|$

compute $p(C_r=1|C_1=c_1, \ldots, C_{r\text{-}1}=c_{r\text{-}1})$ given M and $c_{1 \ldots} c_{r\text{-}1}$

convert $p(C_r=1|C_1=c_1, \ldots)$ to $\{0, 1\}$ using a Bernoulli function

(Chuklin, et al., 2016)

45

45

Click Models

Click models can be used to generate artificial relevance judgments

- Any individual click is a noisy signal
- Use a click model such as DCTR to learn ρ_{uq} for each query-document pair *uq*
- Measure NDCG@k using ρ_{uq} as the relevance label

46

46

Lecture Outline

- Segmenting search logs into sessions
- Query suggestions
- Query intents
- Click models

47 © 2021, Jamie Callan

47

For More Information

- E. Agichtein, E. Brill, S. Dumais, and R. Ragno. "Learning user interaction models for predicting Web search result preferences." *Proceedings of SIGIR 2006*. 2006.
- C. Eickhoff, J. Teevan, R. White, and S. Dumais. "Lessons from the journey: A query log analysis of within-session learning." In Proceedings of WSDM '14. 2014
- T. Joachims, L. Granka, and B. Pan. "Accurately interpreting clickthrough data as implicit feedback." *Proceedings of SIGIR 2005*. 2005.
- A. Chuklin, I. Markov, and M. de Rijke. Click models for web search. Morgan and Claypool. 2016.
- R. Jones and K.L. Klinker. "Beyond the session timeout: Automatic hierarchical segmentation of search topics in query logs." In *Proceedings CIKM '08*. 2008.
- R. Jones, B. Rey, O. Madani, and W. Greiner. "Generating query substitutions." WWW 2006.
- G. Pass, A. Chowdhury, and C. Torgeson. A picture of search. In *InfoScale '06: Proceedings of the 1st International Conference on Scalable Information Systems.* 2006.
- F. Radlinski, M. Szummer, and N. Craswell. "Inferring query intent from reformulations and clicks." WWW 2010.
- R. White, M. Bilenko, and S. Cucerzan. "Studying the use of popular destinations to enhance web search interaction." *Proceedings of SIGIR 2007*. 2007.

48 © 2021, Jamie Callan

48