Information Retrieval: Real-Time Search

Fernando Diaz
Microsoft Research NYC

November 21, 2016
Detecting Emerging Topics in Query Streams
Integration of News Content into Web Search Results

[WSDM 2010]

chile earthquake

web index

news index
Clicks and News
Intuition

- a click on a news display suggests an appropriate presentation
- a skip over a news display suggests an inappropriate presentation
- click-through rate (CTR) summarizes clicks and skips,

\[
\text{CTR} = \frac{\text{clicks}}{\text{clicks} + \text{skips}}
\]
Analyzing Full Recall Data

News Results for Inauguration
- Online inauguration videos set records CNN - 3 hours ago
- Castro watched inauguration, Argentine leader says CNN - 3 hours ago
- Photographer: Inauguration like no moment I've ever witnessed CNN - 4 hours ago

Inauguration Day - Wikipedia
The swearing-in of the President of the United States occurs upon the commencement of a new term of a President of the United States. The United States Constitution mandates that the President make the following oath or...
http://en.wikipedia.org/wiki/United_States_presidential_inauguration

Joint Congressional Committee on Inaugural Ceremonies
Charged with planning and conducting the inaugural activities at the Capitol: the swearing-in ceremony and the luncheon honoring the President and Vice President.
http://inaugural.senate.gov

Inauguration Day 2009
Official site for the 2009 Inauguration of Barack Obama. Provides information about events, tickets, and inaugural balls and parades.
http://inaugural.senate.gov/2009

Inaugural Addresses of the Presidents of the United States
From George Washington's first address in 1789 to the present. Includes a note on the presidents who took the oath of office without a formal inauguration.
http://www.bartleby.com/124

- for a small percentage of search traffic, present a news display if there are any hits in the news index
- collect user click and skip information
- for each query, compute the click-through rate if we always present a display for that query
CTR Distribution

queries binned logarithmically
newsworthy $\rightarrow$ high CTR

- ask annotators to generate a set of newsworthy queries during our data collection
- annotators provided access to,
  - realtime news corpus
  - realtime query logs
  - other media sources (e.g. television, radio)
- average CTR of queries: 0.249
high CTR $\rightarrow$ newsworthy

- select 20 queries from each bin
- ask annotators to label each query on a five point scale between newsworthy and non-newsworthy
Approach

- when a query is submitted, predict its CTR
- if the predicted CTR is above some threshold, present the display
- if the predicted CTR is below some threshold, do not present the display
Estimating CTR
Estimating CTR

- for each query event, we would like to estimate the probability that the display will be clicked
- naïve estimation
  - always present display, gather click and skip evidence
  - $\hat{p}_q^t = \frac{c_q^t}{v_q^t}$
- problems with naïve estimation
  - requires presenting a display for many inappropriate queries
Contextual Evidence for Predicting CTR of Unseen Queries

• when a new query enters the system, we have information which may help predict the CTR,
  • volume in web query log
  • volume in news vertical query log
  • growth in query logs
  • document hits in the last $k$ hours, days
  • mean age of retrieved documents
  • precision predictors (e.g. clarity)
  • distributed IR metrics (e.g. ReDDE)
Predicting CTR Without Click Information

- without click information, we can use contextual features to predict CTR
  - query-independent features allow extension to unseen queries
- model using logistic regression trained on full-recall data
- $\pi^t_q$: predicted CTR for query $q$ at time $t$ using only contextual features
Modeling CTR with a Beta Prior

Assume that the CTR is Beta distributed,

\[ p_t^q \sim \text{Beta}(a, b) \]

We can set the parameters of the prior using our contextual prediction,

\[ a = \mu \pi_t^q \quad b = \mu (1 - \pi_t^q) \]
Prior Distribution over CTR

\[ \mu = 10 \quad \pi^t_q = 0.30 \]
Prior Distribution over CTR

\[ \mu = 100 \quad \pi^t_q = 0.30 \]
Posterior Distribution over CTR

Given $C_q^t$ clicks and $S_q^t$ skips,

$$p_q^t | C_q^t, S_q^t \sim \text{Beta}(a + C_q^t, b + S_q^t)$$

And the posterior mean,

$$\tilde{p}_q^t = \frac{C_q^t + \mu \pi_q^t}{\mathcal{V}_q^t + \mu}$$
Posterior Distribution over CTR

\[ C_q = 0 \quad \pi_q^t = 0.30 \]

\[ S_q = 10 \quad \mu = 10 \]
Posterior Distribution over CTR

\[ C_q = 10 \quad \pi_q^t = 0.30 \]
\[ S_q = 0 \quad \mu = 10 \]
Modification: Exploiting Similar Queries

- evidence can also be provided by topically related queries
- query similarity
  - build relevance model for each query
  - compute Bhattacharyya correlation between relevance models
- incorporate evidence from similar queries as partial pseudo-counts
Classifying Queries
From Probabilities to Decisions

- Present a display if $\tilde{p}_q^t > \tau$
- If clicks indicate relevance, then $\tau = \frac{1}{2}$
- We know from motivating experiments that clicks $\neq$ relevance
- Therefore, we derive our threshold from an accuracy measure.
Accuracy

\[ A = \frac{C^+ + S^+}{C^* + S^*} \]

- \( C^+_q \): correctly predicted clicks
- \( S^+_q \): correctly predicted skips
- \( C^*_q \): total clicks, seen and unseen
- \( S^*_q \): total skips, seen and unseen

Given this formula, we can derive \( \tau = \frac{1}{2} \).
$\alpha$-Accuracy

$$A_\alpha = \frac{\alpha C_q^+ + S_q^+}{\alpha C_q^* + S_q^*}$$

$C_q^+$ correctly predicted clicks
$S_q^+$ correctly predicted skips
$C_q^*$ total clicks, seen and unseen
$S_q^*$ total skips, seen and unseen

where $\alpha \geq 1$ and controls the importance we place on detecting clicks. Given $\alpha$, we can derive $\tau = \frac{1}{\alpha+1}$.
Modification: Opportunistic Sampling

- Sometimes the value of feedback information outweighs a small degradation in performance
- Addresses false negatives predicted by the contextual model
- Naïve method: if $\tilde{p}_q^t < \tau$, then present a display for the first $k$ issuances of a query
Sampling from the Posterior

- Naïve method treats all below threshold queries equally
- The posterior distribution contains confidence information
- Alternative method: if $\hat{p}_q^t < \tau$, then sample a CTR, $\hat{p}_q^t$, from the posterior. If $\hat{p}_q^t > \tau$, then present.
Threshold

\[ \mu = 10 \quad \pi^t_q = 0.15 \]

\[ P(p^t_q > \tau) = 0.272 \quad \tau = 0.20 \]
Threshold

\[ \mu = 100 \quad \pi^t_q = 0.15 \]
\[ P(p^t_q > \tau) = 0.088 \quad \tau = 0.20 \]
Experiments
Experimental Setup

• Data
  • Two full recall datasets gathered in Spring 2007 and Winter 2008.
  • Simulate decision-making with recorded click and skip information

• Evaluation
  • binned macro-averaged (by query) accuracy
Binned, Macro-averaged Accuracy

- evaluation algorithm
  - bin queries by empirical CTR using full recall data
  - compute accuracy ($A_4$) for each query
  - compute the average accuracy for each bin
- these numbers are normalized by performance of an omniscient algorithm
<table>
<thead>
<tr>
<th>run</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>only use the contextual prior</td>
</tr>
<tr>
<td>history</td>
<td>use the prior and click feedback</td>
</tr>
<tr>
<td>similarity</td>
<td>use the prior and click feedback using similar queries</td>
</tr>
<tr>
<td>posterior</td>
<td>use similarity baseline but present by sampling from the posterior</td>
</tr>
</tbody>
</table>
## Binned, Macro-averaged Accuracy

<table>
<thead>
<tr>
<th>baseline</th>
<th>history</th>
<th>similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0.358</td>
<td>0.421</td>
<td>0.615*</td>
</tr>
<tr>
<td>2 0.576</td>
<td>0.636</td>
<td>0.820*</td>
</tr>
<tr>
<td>3 0.598</td>
<td>0.680</td>
<td>0.776*</td>
</tr>
<tr>
<td>4 0.620</td>
<td>0.654</td>
<td>0.738*</td>
</tr>
<tr>
<td>5 0.722</td>
<td>0.731</td>
<td>0.796*</td>
</tr>
<tr>
<td>6 0.901</td>
<td>0.885</td>
<td>0.896*</td>
</tr>
<tr>
<td>7 0.978</td>
<td>0.982</td>
<td>0.979</td>
</tr>
<tr>
<td>8 0.964</td>
<td>0.984</td>
<td>0.977°</td>
</tr>
<tr>
<td>9 0.973</td>
<td>0.991</td>
<td>0.987°</td>
</tr>
<tr>
<td>10 0.985</td>
<td>0.995</td>
<td>0.992°</td>
</tr>
<tr>
<td>all 0.964</td>
<td>0.978</td>
<td>0.978</td>
</tr>
</tbody>
</table>

significant increases over baseline in **bold**; decreases under baseline in *italic*. significant increases over history indicated by *; decreases under history indicated by o.
Binned, Macro-averaged Accuracy

```
baseline

similarity
```

```
10 9 8 7 6 5 4 3 2 1
0.0
0.2
0.4
0.6
0.8
1.0
baseline
10 9 8 7 6 5 4 3 2 1
0.0
0.2
0.4
0.6
0.8
1.0
similarity
```
Sampling from Posterior

![Box plots for similarity and mu=10](image)
Conclusions

- contextual model can be used as a lightweight detector
- feedback information valuable for handling false positives
- sampling valuable for handling false negatives
Real-Time Search
persian food sf

web index

news index
Mis-Handling Queries on Emerging Topics is Catastrophic

- **Guaranteed zero recall**: if no on-topic information exists in the index, the user can never find it no matter the reformulations (without prior knowledge).
- **Information need is urgent**: the user wants information immediately and is willing to switch providers in order to find it.
- **High visibility failure**: like it or not, the media often uses breaking news queries to compare search engines.
Michael Jackson Dead: Microsoft Bing FAILS in Coverage, Twitter and Facebook Break News

Has Michael Jackson died or is this another social media and Internet hoax? Social Media is reporting his death. CNN is reporting he’s in a coma. And the search engines don’t have a clue what’s going on.

Social Media: Twitter Coverage of TMZ

According to Twitter and Facebook users all over the world, Michael Jackson died today from a cardiac arrest after being rushed to the hospital. However, there is no major news network announcing his death, except for one unconfirmed “EXCLUSIVE” on AOL’s TMZ blog.

In addition to TMZ, the only other real sources reporting that Michael Jackson has died are celebrities on Twitter, and Twitter users retweeting the TMZ story and celebrity tweets.

[Search Engine Journal, June 25, 2009]
So How Are The Search Engines Reporting It?

In terms of search relevance and breaking news, even with conflicting news amongst various media outlets and social media, Google has not caught up to the rush of Michael Jackson news. Google is showing only ONE headline in its Google News Universal Search Onebox about the rumored passing of Jackson, with others about his jewelry and one about Lou Ferigno training Mr. Jackson.

[Search Engine Journal, June 25, 2009]
Research Questions

- How to detect that a query is newsworthy?
- How to find new fresh/useful documents to index?
- How to rank fresh documents?
Satisfying Emerging Topics in Query Streams
Sources of Realtime Data for Resource Discovery

- **User monitoring**: toolbars, browsers, DNS requests
- **Interaction monitoring**: email, IM, SMS
- **Realtime personal publishing**: blogs, twitter, delicious
Sources of Realtime Data for Resource Discovery

- **User monitoring**: toolbars, browsers, DNS requests
- **Interaction monitoring**: email, IM, SMS
- **Realtime personal publishing**: blogs, twitter, delicious
Twitter Data

- Tweets include embedded urls.
- Tweets are generated by unique users.
- Tweets include text.
Outline

• Using Twitter for New Resource Discovery
• Using Twitter for New Resource Ranking
• Experiments
Using Twitter for New Resource Discovery
Using Twitter for New Resource Discovery

- Hypothesis: All Twitter URLs should be indexed as posted to allow immediate retrievability.
- Reality: Many Twitter URLs are spam or self-promotion URLs.
- Solution: Apply high-precision heuristic spam filter.
- Future Work: Improve spam filtering.
Using Twitter for New Resource Ranking
Using Twitter for New Resource Ranking

- Naïve Solution: Place in static web index, rank using existing relevance model.
- Reality: Fresh URLs have unique—and often impoverished—ranking features.
- Solution: Leverage Twitter information on fresh URLs.
Digression: Machine-Learned Ranking

Features

\[ d_i = \begin{cases} 
\text{query term matches in title} \\
\text{query term matches in body} \\
\text{popularity} \\
\text{anchor text match} \\
\text{pagerank} \\
\vdots \\
\end{cases} \]

- content features
- aggregate features
Digression: Machine-Learned Ranking

Scoring

\[ f(d_i, M) = y_i \]

- \( M \): model parameters
- \( y_i \): document score
Digression: Machine-Learned Ranking

Training

\[ \text{argmin}_M \sum_{d_i \in D} \ell(f(d_i, M), y_i) \]

D \text{  training data}
Digression: Machine-Learned Ranking

\[ f \left( \begin{bmatrix} d_1^T \\ d_2^T \\ \vdots \\ d_n^T \end{bmatrix}, \mathcal{M} \right) = y \]
Problems with Fresh URLs

- The majority of queries—and therefore training data—uses queries that are not on emerging topics.
- Aggregate features well defined for relevant document.
- Fresh URLs have impoverished/non-existent aggregate features.
- Solution: Leverage Twitter Data for Fresh URLs.
Twitter Features

- Tweet content feature: similarity of anchor tweets to query (e.g. word-level, phrase-level).
- Tweeter features: network authority of anchor Tweeter.
Other Twitter Features

<table>
<thead>
<tr>
<th>$\phi_{\text{other-1}}$</th>
<th>average number of followers for the users who issued the tiny URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\phi_{\text{other-2}}$</td>
<td>average post number for the users who issued the tiny URL</td>
</tr>
<tr>
<td>$\phi_{\text{other-3}}$</td>
<td>average number of users who retweeted the tweets containing the tiny URL</td>
</tr>
<tr>
<td>$\phi_{\text{other-4}}$</td>
<td>average number of users who replied those users that issued the tiny URL</td>
</tr>
<tr>
<td>$\phi_{\text{other-5}}$</td>
<td>average number of followings for the users who issued the tiny URL</td>
</tr>
<tr>
<td>$\phi_{\text{other-6}}$</td>
<td>average Twitter score of all the users who issued the tiny URL</td>
</tr>
<tr>
<td>$\phi_{\text{other-7}}$</td>
<td>number of followers for the user who first issued the tiny URL</td>
</tr>
<tr>
<td>$\phi_{\text{other-8}}$</td>
<td>number of posts by the user who first issued the tiny URL</td>
</tr>
<tr>
<td>$\phi_{\text{other-9}}$</td>
<td>number of users who retweeted the user who first issued the tiny URL</td>
</tr>
<tr>
<td>$\phi_{\text{other-10}}$</td>
<td>number of users who replied the user who first issued the tiny URL</td>
</tr>
<tr>
<td>$\phi_{\text{other-11}}$</td>
<td>number of followings for the user who first issued the tiny URL</td>
</tr>
<tr>
<td>$\phi_{\text{other-12}}$</td>
<td>Twitter score of the users who first issued the tiny URL</td>
</tr>
<tr>
<td>$\phi_{\text{other-13}}$</td>
<td>number of followers for the user who issued the tiny URL with the highest Twitter score</td>
</tr>
<tr>
<td>$\phi_{\text{other-14}}$</td>
<td>number of posts by the user who issued the tiny URL with the highest Twitter score</td>
</tr>
<tr>
<td>$\phi_{\text{other-15}}$</td>
<td>number of users who retweeted the user who issued the tiny URL and has the highest Twitter score</td>
</tr>
<tr>
<td>$\phi_{\text{other-16}}$</td>
<td>number of users who replied the user who issued the tiny URL and has the highest Twitter score</td>
</tr>
<tr>
<td>$\phi_{\text{other-17}}$</td>
<td>number of followings for the user who has the highest Twitter score among the users that issued the tiny URL</td>
</tr>
<tr>
<td>$\phi_{\text{other-18}}$</td>
<td>Twitter score of the users who issued the tiny URL and who is the highest Twitter score</td>
</tr>
<tr>
<td>$\phi_{\text{other-19}}$</td>
<td>number of different users who sent the tiny URL.</td>
</tr>
</tbody>
</table>
\[ d_i = \begin{cases} 
\text{query term matches in title} \\
\text{query term matches in body} \\
\text{popularity} \\
\text{anchor text match} \\
\text{pagerank} \\
\text{Tweet content match} \\
\text{Tweeter authority} \\
\text{Other Twitter features} \\
\end{cases} \]

\{ \text{content features} \}

\{ \text{aggregate features} \}

\{ \text{Twitter features} \}
\[ \mathbf{d}_{\text{old}} = \{ 2, 3, 0.75, 3, 0.01, \ldots \} \]

- **content features**
- **aggregate features**
- **Twitter features**
\[ d_{\text{fresh}} = \begin{cases} 2 \\ 3 \\ 0 \\ 0 \\ 4 \\ 0.90 \\ 1.0 \end{cases} \]
Trained Models

Train-Models($D_{\text{regular}}$, $D_{\text{Twitter}}$)

$D_{\text{regular}}$: training data set from regular data
$D_{\text{Twitter}}$: training data set from Twitter data

1. $M_{\text{regular}} \leftarrow \text{Train-MLR}(D_{\text{regular}}, \{F_{\text{content}}, F_{\text{aggregate}}\})$
2. $M_{\text{Twitter}} \leftarrow \text{Train-MLR}(D_{\text{Twitter}}, \{F_{\text{content}}, F_{\text{Twitter}}\})$
3. $M_{\text{content}} \leftarrow \text{Train-MLR}(D_{\text{regular}}, F_{\text{content}})$
## Runs

| (\( M_{\text{regular}}; M_{\text{regular}} \)) | Use \( M_{\text{regular}} \) on regular and Twitter URLs. |
| (\( M_{\text{content}}; M_{\text{content}} \)) | Use \( M_{\text{content}} \) on regular and Twitter URLs. |
| (\( M_{\text{regular}}; M_{\text{content}} \)) | Use \( M_{\text{regular}} \) on regular URLs and \( M_{\text{content}} \) on Twitter URLs. |
| (\( M_{\text{regular}}; M_{\text{twitter}} \)) | Use \( M_{\text{regular}} \) on regular URLs and \( M_{\text{twitter}} \) on Twitter URLs. |
| (\( M_{\text{regular}}; M_{\text{composite}} \)) | Use \( M_{\text{regular}} \) on regular URLs and \( M_{\text{composite}} \) on Twitter URLs. |
Experiments
Data

- Editorial judgments on breaking news queries (automatically classified).
  - Labels included both relevance grade and freshness.
  - **Training**: 206,249 query-document pairs; 5006 query-Twitter document pairs.
  - **Testing**: 3781 query-document pairs; 769 query-Twitter document pairs.
- Performance measured according to DCG.
- All improvements significant using t-test ($p < 0.01$).
**Metrics**

\[ DCG_n^g = \sum_{i=1}^{n} \frac{g(i)}{\log_2(i + 1)} \]

\( g(i) \) computes the gain of document \( i \)
(relevance, freshness, etc.)
Gain Functions

\[ g(i) = r(i) \quad \text{relevance} \]
\[ g(i) = f(i) \quad \text{freshness} \]
\[ g(i) = c(i) \quad \text{combined} \]
\[ = \max(r(i) - (f^* - f(i)), 0) \]
### Results

<table>
<thead>
<tr>
<th>Model Configuration</th>
<th>NDCG(^c)_1</th>
<th>NDCG(^r)_1</th>
<th>NDCG(^f)_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>((M_{\text{regular}}), (M_{\text{regular}}))</td>
<td>0.588</td>
<td>0.611</td>
<td>0.474</td>
</tr>
<tr>
<td>((M_{\text{content}}), (M_{\text{content}}))</td>
<td>0.570</td>
<td>0.610</td>
<td>0.513</td>
</tr>
<tr>
<td>((M_{\text{regular}}), (M_{\text{content}}))</td>
<td>0.600</td>
<td>0.618</td>
<td>0.520</td>
</tr>
<tr>
<td>((M_{\text{regular}}), (M_{\text{twitter}}))</td>
<td>0.720</td>
<td>0.708</td>
<td>0.717</td>
</tr>
</tbody>
</table>

- Use \(M_{\text{regular}}\) on regular and Twitter URLs.
- Use \(M_{\text{content}}\) on regular and Twitter URLs.
- Use \(M_{\text{regular}}\) on regular URLs and \(M_{\text{content}}\) on Twitter URLs.
- Use \(M_{\text{regular}}\) on regular URLs and \(M_{\text{twitter}}\) on Twitter URLs.
## Results

<table>
<thead>
<tr>
<th>Configuration</th>
<th>(\text{NDCG}_5^c)</th>
<th>(\text{NDCG}_5^r)</th>
<th>(\text{NDCG}_5^f)</th>
</tr>
</thead>
<tbody>
<tr>
<td>((\mathcal{M}<em>{\text{regular}}, \mathcal{M}</em>{\text{regular}}))</td>
<td>0.666</td>
<td>0.681</td>
<td>0.518</td>
</tr>
<tr>
<td>((\mathcal{M}<em>{\text{content}}, \mathcal{M}</em>{\text{content}}))</td>
<td>0.652</td>
<td>0.682</td>
<td>0.587</td>
</tr>
<tr>
<td>((\mathcal{M}<em>{\text{regular}}, \mathcal{M}</em>{\text{content}}))</td>
<td>0.680</td>
<td>0.690</td>
<td>0.569</td>
</tr>
<tr>
<td>((\mathcal{M}<em>{\text{regular}}, \mathcal{M}</em>{\text{twitter}}))</td>
<td><strong>0.739</strong></td>
<td><strong>0.729</strong></td>
<td><strong>0.736</strong></td>
</tr>
</tbody>
</table>

- Use \(\mathcal{M}_{\text{regular}}\) on regular and Twitter URLs.
- Use \(\mathcal{M}_{\text{content}}\) on regular and Twitter URLs.
- Use \(\mathcal{M}_{\text{regular}}\) on regular URLs and \(\mathcal{M}_{\text{content}}\) on Twitter URLs.
- Use \(\mathcal{M}_{\text{regular}}\) on regular URLs and \(\mathcal{M}_{\text{twitter}}\) on Twitter URLs.
Conclusions

- Emerging news queries demand unique query triage, crawling, and ranking choices.
- Exploit user feedback for query detection.
- Exploit user behavior for crawling and ranking fresh URLs.