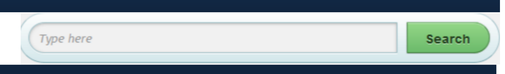
Leveraging Site Search Logs to Identify Missing Content on Enterprise Webpages

Harsh Jhamtani[1], Rishiraj Saha Roy[2], Niyati Chhaya[3], Eric Nyberg[1]
[1]Carnegie Mellon University, [2]Max Plank Institute for Informatics, [3]Adobe Research



Problem Statement

Background:
- Scope for (semi-) automatic maintenance of content on web-pages by inferring user requirements
- Ubiquity of site search box on enterprise websites

Site search box usage scenarios:
- Scenario A: I want to navigate to page about 'X'. Navigating through menus and options is a pain. Let me use site search box.
- Scenario B**:** I expected info about 'X' on this page. It's not there. Let me use site search box. ~ Missing content

Query	% count
Flash download	6.9
Illustrator pricing	5.9
Illustrator download	5.0

Table 1: Percentage count of few queries for referral webpage: http://www.adobe.com/products/illustrator.html

Challenges:
- Disambiguating missing content scenario from navigational scenario. Naively using query counts will give misleading results
- Coming up with methods to help in rectification of missing content issues.

Notation:
A query *q* issued from referral webpage *w* is represented through tuple (w,q). The website i.e. the collection of webpages is represented by *W*

Method

Phase 1 (Figure 1)
- We make the following assumption for *navigation* scenario: Distribution of queries will be independent of referral webpage i.e. P(Q=q|w) = P(Q=q).
- We calculate deviation from this behavior using pearson residuals. (w,q) tuples with $e_{ij} > \delta$ are *missing content tuples*.

$$\mu_{ij} = p_{i+} \times p_{+j} \times M \qquad e_{ij} = \frac{C[i][j] - \mu_{ij}}{\sqrt{\mu_{ij}(1-p_{i+})(1-p_{+j})}}$$

Phase 2 (Figure 2)

Consider following two measures for a tuple (w*,q*)
1. page se score (w*, q*): Relevance score of page w* for query q*, as provided by the site search engine
2. best match score (q* , W): The score of the best matching page in W for q * , again provided by the site search engine

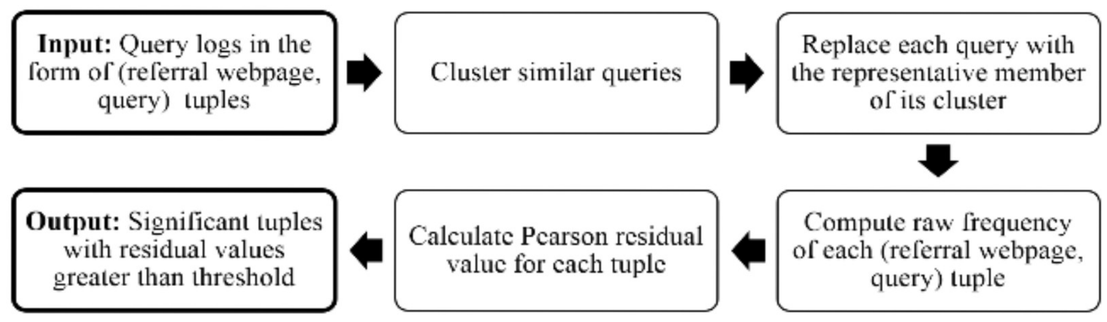


Figure 1: Steps in Phase 1

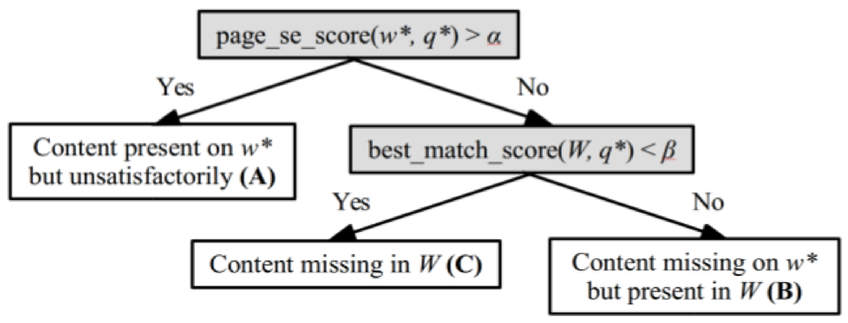


Figure 2: Classification of significant tuples in phase 2

Phase 3: Rectifying issues:

- Missing content on page: Leverage click-through data to infer which search result link was satisfactory to user. Use content / link to the inferred webpage(s)
- Missing content on site: Topics for author to write about
- Unsatisfactorily present content: Leverage click-through data as described earlier

Data

Following data sources are needed for the system:
- Query logs
- Website content
- Click-through data

Item	Count
(w,q) tuples	153K
Distinct queries after clustering	12K
Distinct referral webpage	2K
Distinct (w,q) tuples	26K

Table 2: Description of adobe dataset

Experiments & Results

Parameter	Method
α, β	Using 1000 binary-relevance judged (w, q) pairs (by humans, 500 relevant and non-relevant pairs each).
δ	Positive residuals ~exp(rate = 0.0139). The log likelihood of the fit, normalized by the number of values, was −5.28. We set δ as the mean of the distribution, which was 71.94.
ɣ	0.7 (through manual inspection)

Table 3: Parameter tuning

Class	Count	%
Not statistically significant	18, 639	69.74
Unsatisfactorily present content (A)	580	2.17
Missing content on page (B)	4, 302	16.09
Missing content on site (C)	3, 206	11.99

Table 4: Results on Adobe dataset: Distribution of missing content tuples after classification

Referral webpage	Query	Class
www.adobe.com/	photoshop	Insignificant
www.adobe.com/products/cs6/faq.html	education discount cs	A
www.adobe.com/support/downloads/help.html	removing acrobat 8.0	B
helpx.adobe.com/premiere-pro/topics.html	import not responding	C

Table 5: Examples from various classes for Adobe dataset

Can simple sorting by query counts be used instead ?

- Pearson's rank correlation coefficient (r) between the vectors of counts and residual values over all tuples was found to be very close to zero (−0.035).
- Kendall rank correlation coefficient τ between the ranked lists when (w, q) tuples are ordered by frequency and residual value, was found to be -4.65×10^{-9} .

This indicates almost no correlation between counts and residuals.

Conclusion

We have formulated a practical and novel research problem. Our method is light weight and builds on query logs, which are often readily available.

Closest prior work is of Yom-Tov et al. [1], who try to predict *query difficulty*. However, they work with a collection of documents rather that an enterprises setting.

Future Directions:
Evaluation in a deployed scenario
There is lot of image data in modern websites - that has not been considered.

References:
[1] Yom-Tov, E., Fine, S., Carmel, D., Darlow, A.: Learning to estimate query difficulty including applications to missing content detection and distributed information retrieval. In: SIGIR '05 (2005)

MAX-PLANCK-GESELLSCHAFT

ECIR 2017