Query Intelligence in E-Commerce Search

01 Introduction

Aim

Search Query Intelligence has many flavours of preprocessing like tokenization, vector search etc.

We mainly focus on User clicks data and how it can improve the existing search functionality. (Feedback loops)

Our idea

- Apache Solr (for searching)
- Apache Spark (for preprocessing)

Based on the user clicks data, do preprocessing, and improve the accuracy of the boost values in the search query.

```
{
  "query":["keyword1"^31,"keyword2"^43,"keyword3"^82,"keyword4"^31]
  "fields": ["upc"^42, "name"^45, "manufacturer"^49, "score"^52],
  "limit": 5,
  "params": {
    "qf": "name manufacturer longDescription",
    "defType": "edismax",
    "indent": "true",
    "sort": "score desc, upc asc",
    "qf": "name manufacturer longDescription",
    "boost": "sum(1,query({! df=upc v=$signals_boosting}))",
    "signals_boosting": ["885909457588"^966, "885909457595"^205,
"885909471812"^202,"886111287055"^109]
  }
```

02 Datasets

- Products.csv

  48195 rows,
  5 columns (doc_id, name,manufacturer, short Description, long Description)

- Signals.csv

  2M rows
  5 columns (session_id,user_id, type, target, time)

  If the type is query, then the target is query text that user has entered.

  If the type is click, add-to-cart,purchase then target is doc_id of product.csv

TLDR:- Products.csv and Signals.csv are interlinked.

https://github.com/ai-powered-search/retrotech

Boosting products for a query

```
"885909457588"^966 "885909457595"^205 "885909471812"^202 "886111287055"^109 "843404073153"^73
"885909457601"^62 "635753493559"^62 "885909472376"^61 "610839379408"^29 "884962753071"^28
```

- Basic Intuition:-
  From User clicks data, for each query see what products have been clicked and how many times and try to store them in a separate collection.

- Next time, when user enters any query, search in that collection and identify the document and their frequency (boost) and form a list and search in original query

- So, basically we are recommending products with more clicks.

Finding Related Keywords

k1	k2	n_users1	n_users2	pmi2	comp_score
ipad	hp touchpad	7554	4829	1.2318940540272372	1.0
ipad 2	ipad	2842	7554	1.430517155037946	1.25
tablet	ipad	1818	7554	1.6685364924472557	1.6666666666666667
touchpad	ipad	2785	7554	1.2231908670315748	2.125
tablets	ipad	1627	7554	1.7493143317791537	2.6
ipad2	ipad	1254	7554	1.9027023623302282	3.0833333333333335
ipad	apple	7554	1814	1.4995901756327583	3.5714285714285716
touchpad	hp touchpad	2785	4829	1.3943192464710108	4.0625
ipad	hp tablet	7554	1421	1.5940745096856273	4.555555555555555
ipod touch	ipad	2931	7554	0.8634782989267505	5.05
ipad	i pad	7554	612	2.415162433949984	5.545454545454546
kindle	ipad	2833	7554	0.827835342752348	6.041666666666667
laptop	ipad	3554	7554	0.5933664189857986	6.538461538461538
ipad	apple ipad	7554	326	2.916383652644603	7.035714285714286
ipad 2	hp touchpad	2842	4829	1.1805849845414993	7.533333333333333
laptops	laptop	3369	3554	1.2902371152378296	8.03125
...					
ipad	i pad 2	7554	204	3.180197301966425	10.025

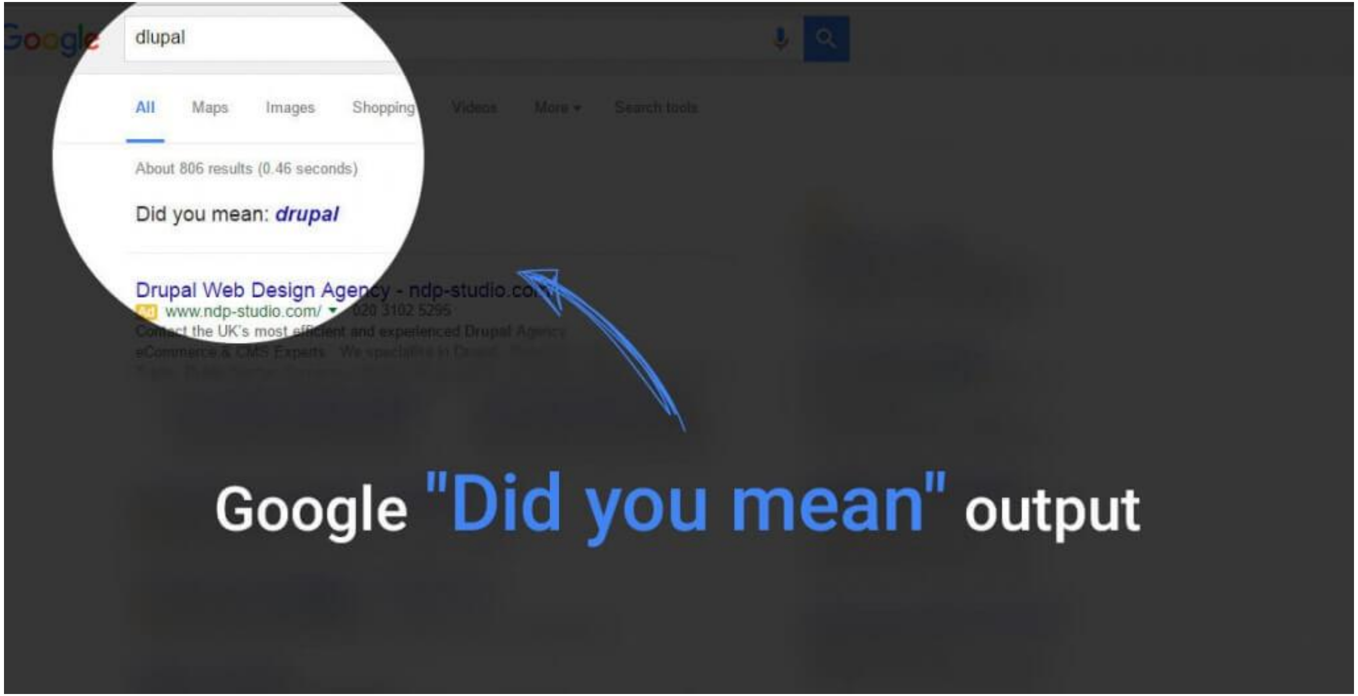
Basic Intuition:-

If user_1 types a query_1 and clicks on a product p

If user_2 types a query_2 and clicks on same product p

query_1 and query_2 are semantically similar

We use PMI and composite score statistics to balance n_users1 and n_users2


dlupal
All
Maps
Images
Shopping
About 806 results (0.46 seconds)
Did you mean: drupal
Drupal Web Design Agency - ndp-studio.com
www.ndp-studio.com/
020 3102 5295
Google "Did you mean" output

Finding Misspelled Words

	misspell	correction	misspell_counts	correction_counts	edit_dist
50	iphone3	iphone	6	16854	1
61	laptopa	laptop	6	14119	1
62	latop	laptop	5	14119	1
136	toucpad	touchpad	6	11550	1
137	touxhpad	touchpad	5	11550	1
148	wirless	wireless	6	10060	1
127	tableta	tablet	6	8260	1
10	cape	case	5	7541	1
8	cage	case	6	7541	1
30	gallaxy	galaxy	6	5839	1
64	loptops	laptops	5	5565	1
90	potable	portable	6	4477	1
5	bluetooh	bluetooth	5	4461	1
146	wats	wars	5	4179	1
56	kimdle	kindle	5	4129	1
99	rauter	router	5	4067	1
77	modum	modem	6	3590	1
76	moden	modem	5	3590	1
135	tosheba	toshiba	6	3432	1
34	gates	games	6	3239	1

Basic Intuition:-

Calculate the frequency for each query.

Quantile (0-20%) would likely be a misspelled word as less frequently used.

(80-100%) quantile would be a correct word as more frequently used.

Find the relation between (0-20%) and (80-100%) using edit distance.

Generalizable search systems

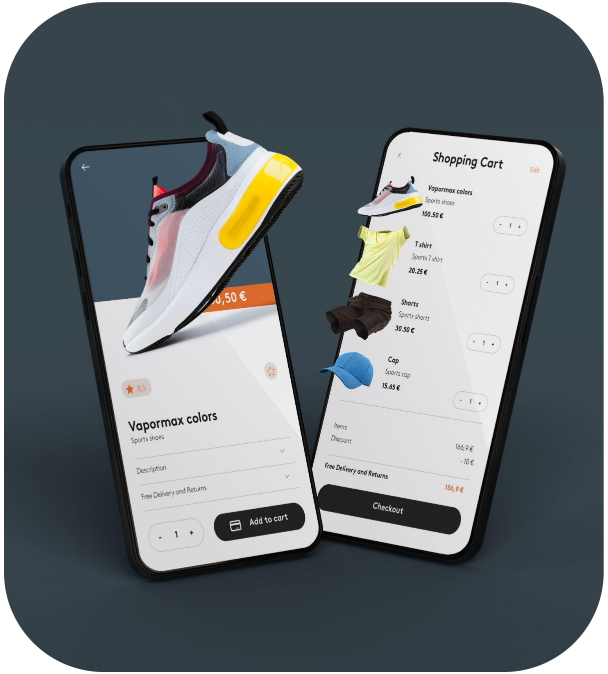
```
q=title:(${keywords})^10 overview:(${keywords})^20 {!func}release_year^0.01
```

- Above is a manual ranking function combining title, overview, and release year weights
- as searched keywords
- Learning to Rank (LTR) takes our proposed relevance factors, and learns an optimal ranking function and brings relevant documents to the top, and push irrelevant ones to the bottom.
- We'll find the optimal weights for title, overview, and release_year in a scoring function like the one above.

Bias in our data

It turns out that these click data have biases. These include:

- Position bias - Position bias is present in of most search systems. If users are shown search results, they tend to prefer highly ranked search results over lower ones - even when those lower results are in fact more relevant.
- Confidence bias - Documents with little click data influence the judgments the same as documents with more click data.



Addressing the bias in our data

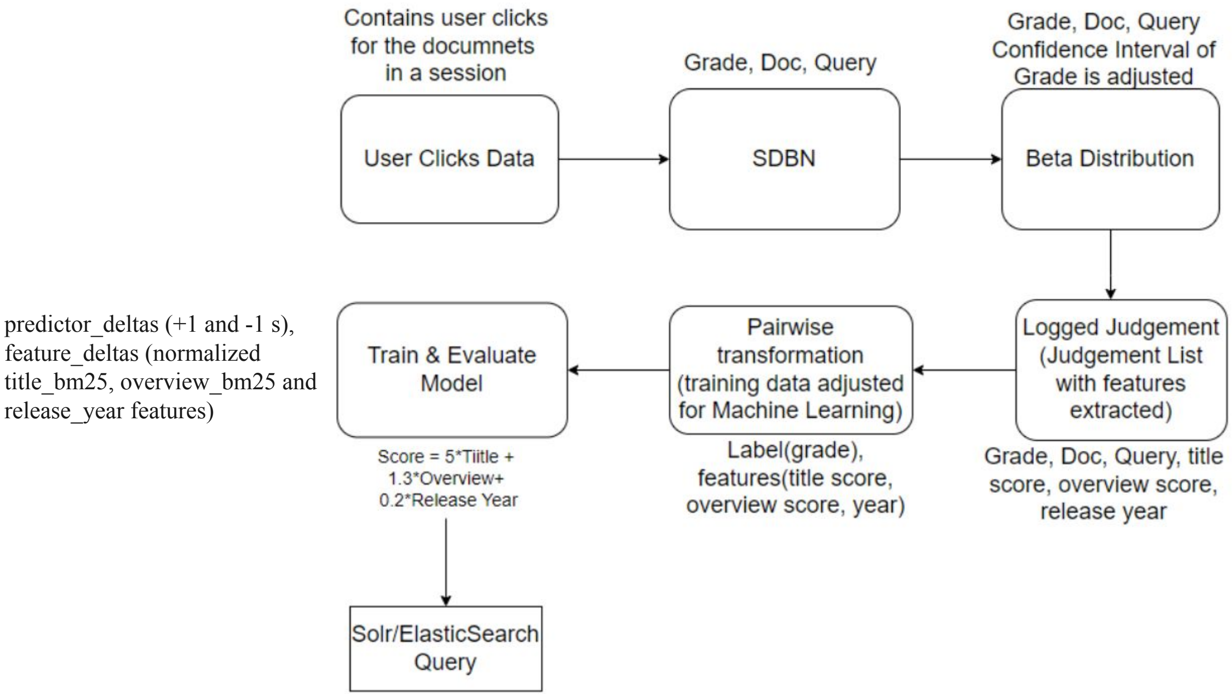
Position Bias

- Detecting Position Bias? If the top positions have a significantly higher CTR
- Addressing Position Bias with SDBN:
  - **Mark the Last Click** - In each search session, identify the last result that was clicked.
  - **Consider Examinations** - Assume that all results above (and including) the last clicked result were actually seen or "examined" by the user.
  - **Calculate Relevance (Grade)**: No. of clicks/No. of examines.

Confidence bias

- Detecting Confidence bias? Document gets clicked every time it's examined, but it's only been examined a few times
- Addressing Position Bias with Beta Distribution
  - **Setting up the Beta Distribution -** Setting a prior belief about document relevance using a default grade and weight, which establish initial values for a (prior_a) and b (prior_b) i.e. (prior_grade = prior_a / (prior_a + prior_b))
  - **Updating with Click Data -** As clicks and examines are observed, "a" is incremented by the click count and "b" by the count of non-clicked examines
  - **Calculating the Updated Grade -** posterior_a / posterior_a + posterior_b

Proposed Methodology



Evaluation Metrics of Model

- Accuracy
- Precision
- Recall
- F1 score
- Mean Squared Error

Thanks!