Jigs and Lures: Associating Web Queries with Structured Entities

Patrick Pantel
Microsoft Research
Redmond, WA, USA
ppantel@microsoft.com

Ariel Fuxman
Microsoft Research
Mountain View, CA, USA
arielf@microsoft.com

Abstract

We propose methods for estimating the probability that an entity from an entity database is associated with a web search query. Association is modeled using a query entity click graph, blending general query click logs with vertical query click logs. Smoothing techniques are proposed to address the inherent data sparsity in such graphs, including interpolation using a query synonymy model. A large-scale empirical analysis of the smoothing techniques, over a 2-year click graph collected from a commercial search engine, shows significant reductions in modeling error. The association models are then applied to the task of recommending products to web queries, by annotating queries with products from a large catalog and then mining query-product associations through web search session analysis. Experimental analysis shows that our smoothing techniques improve coverage while keeping precision stable, and overall, that our top-performing model affects 9% of general web queries with 94% precision.

1 Introduction

Commercial search engines use query associations in a variety of ways, including the recommendation of related queries in Bing, 'something different' in Google, and 'also try' and related concepts in Yahoo. Mining techniques to extract such query associations generally fall into four categories: (a) clustering queries by their co-clicked url patterns (Wen et al., 2001; Baeza-Yates et al., 2004); (b) leveraging co-occurrences of sequential queries in web search query sessions (Zhang and Nasraoui, 2006; Boldi et al., 2009); (c) pattern-based extraction over lexico-syntactic structures of individual queries (Paşca and Durme, 2008; Jain and Pantel, 2009); and (d) distributional similarity techniques over news or web corpora (Agirre et al., 2009; Pantel et al., 2009). These techniques operate at the surface level, associating one surface context (e.g., queries) to another.

In this paper, we focus instead on associating surface contexts with entities that refer to a particular entry in a knowledge base such as Freebase, IMDB, Amazon's product catalog, or The Library of Congress. Whereas the former models might associate the string "*Ronaldinho*" with the strings "*AC Milan*" or "*Lionel Messi*", our goal is to associate "*Ronaldinho*" with, for example, the Wikipedia entity page "*wiki/AC_Milan*" or the Freebase entity "*en/lionel_mess*". Or for the query string "*ice fishing*", we aim to recommend products in a commercial catalog, such as jigs or lures.

The benefits and potential applications are large. By knowing the entity identifiers associated with a query (instead of strings), one can greatly improve both the presentation of search results as well as the click-through experience. For example, consider when the associated entity is a product. Not only can we present the product name to the web user, but we can also display the image, price, and reviews associated with the entity identifier. Once the entity is clicked, instead of issuing a simple web search query, we can now directly show a product page for the exact product; or we can even perform actions directly on the entity, such as buying the entity on Amazon.com, retrieving the product's oper-

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 83–92,
Portland, Oregon, June 19-24, 2011.

ating manual, or even polling your social network for friends that own the product. This is a big step towards a richer semantic search experience.

In this paper, we define the association between a query string q and an entity id e as the probability that e is relevant given the query q, $P(e|q)$. Following Baeza-Yates et al. (2004), we model relevance as the likelihood that a user would click on e given q, events which can be observed in large query-click graphs. Due to the extreme sparsity of query click graphs (Baeza-Yates, 2004), we propose several smoothing models that extend the click graph with query synonyms and then use the synonym click probabilities as a background model. We demonstrate the effectiveness of our smoothing models, via a large-scale empirical study over real-world data, which significantly reduce model errors. We further apply our models to the task of query-product recommendation. Queries in session logs are annotated using our association probabilities and recommendations are obtained by modeling session-level query-product co-occurrences in the annotated sessions. Finally, we demonstrate that our models affect 9% of general web queries with 94% recommendation precision.

2 Related Work

We introduce a novel application of significant commercial value: entity recommendations for general Web queries. This is different from the vast body of work on query suggestions (Baeza-Yates et al., 2004; Fuxman et al., 2008; Mei et al., 2008b; Zhang and Nasraoui, 2006; Craswell and Szummer, 2007; Jagabathula et al., 2011), because our suggestions are actual entities (as opposed to queries or documents). There is also a rich literature on recommendation systems (Sarwar et al., 2001), including successful commercial systems such as the Amazon product recommendation system (Linden et al., 2003) and the Netflix movie recommendation system (Bell et al., 2007). However, these are entity-to-entity recommendations systems. For example, Netflix recommends movies based on previously seen movies (i.e., entities). Furthermore, these systems have access to previous transactions (i.e., actual movie rentals or product purchases), whereas our recommendation system leverages a different resource, namely query sessions.

In principle, one could consider vertical search engines (Nie et al., 2007) as a mechanism for associating queries to entities. For example, if we type the query "canon eos digital camera" on a commerce search engine such as Bing Shopping or Google Products, we get a listing of digital camera entities that satisfy our query. However, vertical search engines are essentially rankers that given a query, return a sorted list of (pointers to) entities that are related to the query. That is, they do not expose actual association scores, which is a key contribution of our work, nor do they operate on general search queries.

Our smoothing methods for estimating association probabilities are related to techniques developed by the NLP and speech communities to smooth *n*-gram probabilities in language modeling. The simplest are discounting methods, such as *additive smoothing* (Lidstone, 1920) and *Good-Turing* (Good, 1953). Other methods leverage lower-order background models for low-frequency events, such as Katz' backoff smoothing (Katz, 1987), Witten-Bell discounting (Witten and Bell, 1991), Jelinek-Mercer interpolation (Jelinek and Mercer, 1980), and Kneser-Ney (Kneser and Ney, 1995).

In the information retrieval community, Ponte and Croft (1998) are credited for accelerating the use of language models. Initial proposals were based on learning *global* smoothing models, where the smoothing of a word would be independent of the document that the word belongs to (Zhai and Lafferty, 2001). More recently, a number of *local* smoothing models have been proposed (Liu and Croft, 2004; Kurland and Lee, 2004; Tao et al., 2006). Unlike global models, local models leverage relationships between documents in a corpus. In particular, they rely on a graph structure that represents document similarity. Intuitively, the smoothing of a word in a document is influenced by the smoothing of the word in similar documents. For a complete survey of these methods and a general optimization framework that encompasses all previous proposals, please see the work of Mei, Zhang et al. (2008a). All the work on local smoothing models has been applied to the prediction of priors for words in documents. To the best of our knowledge, we are the first to establish that query-click graphs can be used to create accurate models of query-entity associations.

3 Association Model

Task Definition: Consider a collection of entities E. Given a search query q, our task is to compute $P(e|q)$, the probability that an entity e is relevant to q, for all $e \in E$.

We limit our model to sets of entities that can be accessed through urls on the web, such as Amazon.com products, IMDB movies, Wikipedia entities, and Yelp points of interest.

Following Baeza-Yates et al. (2004), we model relevance as the click probability of an entity given a query, which we can observe from click logs of vertical search engines, i.e., domain-specific search engines such as the product search engine at Amazon, the local search engine at Yelp, or the travel search engine at Bing Travel. Clicked results in a vertical search engine are edges between queries and entities e in the vertical's knowledge base. General search query click logs, which capture direct user intent signals, have shown significant improvements when used for web search ranking (Agichtein et al., 2006). Unlike for general search engines, vertical search engines have typically much less traffic resulting in extremely sparse click logs.

In this section, we define a graph structure for recording click information and we propose several models for estimating $P(e|q)$ using the graph.

3.1 Query Entity Click Graph

We define a *query entity click graph*, $QEC(Q \cup U \cup E, C_u \cup C_e)$, as a tripartite graph consisting of a set of query nodes Q, url nodes U, entity nodes E, and weighted edges C_u exclusively between nodes of Q and nodes of U, as well as weighted edges C_e exclusively between nodes of Q and nodes of E. Each edge in C_u and C_e represents the number of clicks observed between query-url pairs and query-entity pairs, respectively. Let $w_u(q,u)$ be the click weight of the edges in C_u, and $w_e(q,e)$ be the click weight of the edges in C_e.

If C_e is very large, then we can model the association probability, $P(e|q)$, as the maximum likelihood estimation (MLE) of observing clicks on e given the query q:

$$\hat{P}_{mle}(e|q) = \frac{w_e(q,e)}{\sum_{e' \in E} w_e(q,e')} \qquad (3.1)$$

Figure 1 illustrates an example query entity graph linking general web queries to entities in a large commercial product catalog. Figure 1a illustrates eight queries in Q with their observed clicks (solid lines) with products in E[1]. Some probability estimates, assigned by Equation 3.1, include: $\hat{P}_{mle}(\text{panfish jigs}, e_1) = 0$, $\hat{P}_{mle}(\text{ice jigs}, e_1) = 1$, and $\hat{P}_{mle}(\text{ice auger}, e_4) = \frac{c_e(\text{ice auger}, e_4)}{c_e(\text{ice auger}, e_3) + c_e(\text{ice auger}, e_4)}$.

Even for the largest search engines, query click logs are extremely sparse, and smoothing techniques are necessary (Craswell and Szummer, 2007; Gao et al., 2009). By considering only C_e, those clicked urls that map to our entity collection E, the sparsity situation is even more dire. The sparsity of the graph comes in two forms: a) there are many queries for which an entity is relevant that will never be seen in the click logs (e.g., “panfish jig” in Figure 1a); and b) the query-click distribution is Zipfian and most observed edges will have very low click counts yielding unreliable statistics. In the following subsections, we present a method to expand QEC with unseen queries that are associated with entities in E. Then we propose smoothing methods for leveraging a background model over the expanded click graph.

Throughout our models, we make the simplifying assumption that the knowledge base E is complete.

3.2 Graph Expansion

Following Gao et al. (2009), we address the sparsity of edges in C_e by inferring new edges through traversing the query-url click subgraph, $UC(Q \cup U, C_u)$, *which contains many more edges than* C_e. If two queries q_i and q_j are synonyms or near synonyms[2], then we expect their click patterns to be similar.

We define the synonymy similarity, $s(q_i, q_j)$ as the cosine of the angle between $\mathbf{q_i}$ and $\mathbf{q_j}$, the click pattern vectors of q_i and q_j, respectively:

$$\text{cosine}(\mathbf{q_i}, \mathbf{q_j}) = \frac{\mathbf{q_i} \cdot \mathbf{q_j}}{\sqrt{\mathbf{q_i} \cdot \mathbf{q_i}} \cdot \sqrt{\mathbf{q_j} \cdot \mathbf{q_j}}}$$

where $\mathbf{q}$ is an n_u dimensional vector consisting of the pointwise mutual information between q and each url u in U, $\text{pmi}(q,u)$:

[1] Clicks are collected from a commerce vertical search engine described in Section 5.1.

[2] A query q_i is a near synonym of a query q_j if most relevant results of q_i are also relevant to q_j. Section 5.2.1 describes our adopted metric for near synonymy.

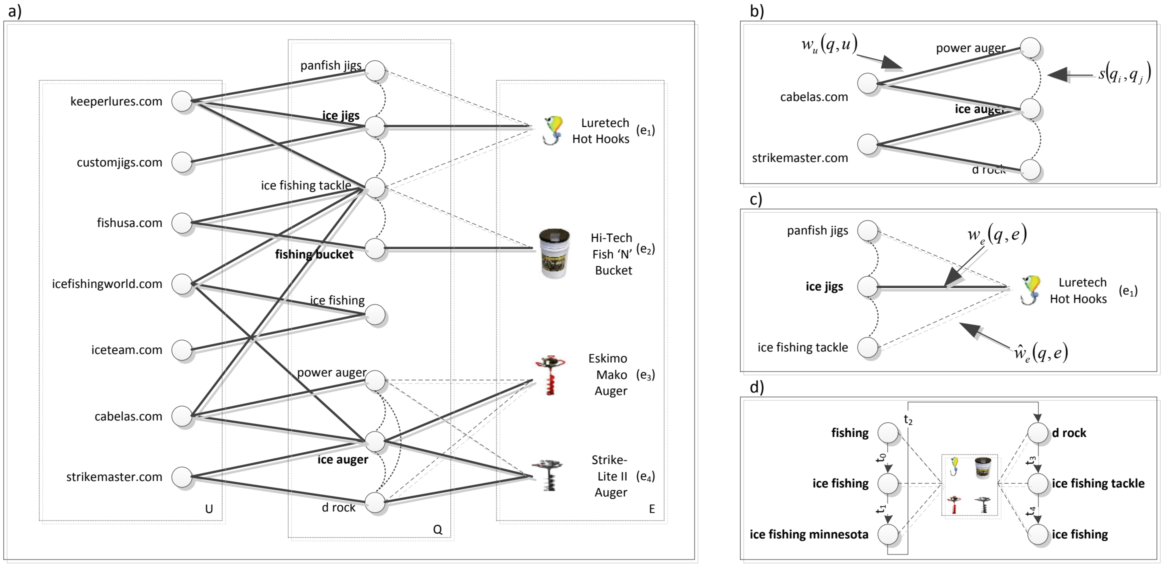

Figure 1: Example QEC graph: (a) Sample queries in Q, clicks connecting queries with urls in U, and clicks to entities in E; (b) Zoom on edges in C_u illustrating clicks observed on urls with weight $w_u(q,u)$ as well as synonymy edges between queries with similarity score $s(q_i,q_j)$ (Section 3.2); (c) Zoom on edges in C_e where solid lines indicate observed clicks with weight $w_e(q,e)$ and dotted lines indicate inferred clicks with smoothed weight $\hat{w}_e(q,e)$ (Section 3.3); and (d) A temporal sequence of queries in a search session illustrating entity associations propagating from the QEC graph to the queries in the session (Section 4).

$$\mathrm{pmi}(q,u) = \log\left(\frac{w_u(q,u)\times\sum_{q'\in Q,u'\in U} w_u(q',u')}{\sum_{u'\in U} w_u(q,u')\sum_{q'\in Q} w_u(q',u)}\right) \tag{3.2}$$

PMI is known to be biased towards infrequent events. We apply the discounting factor, $\delta(q,u)$, proposed in (Pantel and Lin, 2002):

$$\delta(q,u)=\frac{w_u(q,u)}{w_u(q,u)+1}\cdot\frac{\min\left(\sum_{q'\in Q} w_u(q',u),\sum_{u'\in U} w_u(q,u')\right)}{\min\left(\sum_{q'\in Q} w_u(q',u),\sum_{u'\in U} w_u(q,u')\right)+1}$$

Enrichment: We enrich the original QEC graph by creating a new edge $\{q',e\}$, where $q'\in Q$ and $e\in E$, if there exists a query q where $s(q,q')>\rho$ and $w_e(q,e)>0$. ρ is set experimentally, as described in Section 5.2.

Figure 1b illustrates similarity edges created between query "ice auger" and both "power auger" and "d rock". Since "ice auger" was connected to entities e_3 and e_4 in the original QEC, our expansion model creates new edges in C_e between {power auger, e_3}, {power auger, e_4}, and {d rock, e_3}.

For each newly added edge $\{q,e\}$, $\hat{P}_{mle}=0$ according to our model from Equation 3.1 since we have never observed any clicks between q and e. Instead, we define a new model that uses $\hat{P}_{mle}$ when clicks are observed and otherwise assigns uniform probability mass, as:

$$\hat{P}_{hybr}(e|q) = \begin{cases} \hat{P}_{mle}(e|q) & \text{if } \exists e' | w_e(q,e')>0 \\ \frac{1}{\sum_{e'\in E}\phi(q,e')} & \text{otherwise} \end{cases} \tag{3.3}$$

where $\phi(q,e)$ is an indicator variable which is 1 if there is an edge between $\{q,e\}$ in C_e.

This model does not leverage the local synonymy graph in order to transfer edge weight to unseen edges. In the next section, we investigate smoothing techniques for achieving this.

3.3 Smoothing

Smoothing techniques can be useful to alleviate data sparsity problems common in statistical models. In practice, methods that leverage a background model (e.g., a lower-order *n*-gram model) have shown most promise (Katz, 1987; Witten and Bell, 1991; Jelinek and Mercer, 1980; Kneser and Ney, 1995). In this section, we present two smoothing methods, derived from Jelinek-Mercer interpolation (Jelinek and Mercer, 1980), for estimating the target association probability $P(e|q)$.

Figure 1c highlights two edges, illustrated with dashed lines, inserted into C_e during the graph expansion phase of Section 3.2. $\hat{w}_e(q,e)$ represents the weight of our background model, which can be viewed as smoothed click counts, and are obtained

Label	Model	Reference
UNIF	$\hat{P}_{unif}(e\|q)$	Eq. 3.8
MLE	$\hat{P}_{mle}(e\|q)$	Eq. 3.1
HYBR	$\hat{P}_{hybr}(e\|q)$	Eq. 3.3
INTU	$\hat{P}_{intu}(e\|q)$	Eq. 3.6
INTP	$\hat{P}_{intp}(e\|q)$	Eq. 3.7

Table 1: Models for estimating the association probability $P(e|q)$.

by propagating clicks to unseen edges using the synonymy model as follows:

$$\hat{w}_e(q,e) = \sum_{q' \in Q} \frac{s(q,q')}{N_{s_q}} \times \hat{P}_{mle}(e|q') \quad (3.4)$$

where $N_{s_q} = \sum_{q' \in Q} s(q,q')$. By normalizing the smoothed weights, we obtain our background model, $\hat{P}_{bsim}$:

$$\hat{P}_{bsim}(e|q) = \frac{\hat{w}_e(q,e)}{\sum_{e' \in E} \hat{w}_e(q,e')} \quad (3.5)$$

Below we propose two models for interpolating our foreground model from Equation 3.1 with the background model from Equation 3.5.

Basic Interpolation: This smoothing model, $\hat{P}_{intu}(e|q)$, linearly combines our foreground and background models using a model parameter α:

$$\hat{P}_{intu}(e|q) = \alpha \hat{P}_{mle}(e|q) + (1-\alpha)\hat{P}_{bsim}(e|q) \quad (3.6)$$

Bucket Interpolation: Intuitively, edges $\{q,e\} \in C_e$ with higher observed clicks, $w_e(q,e)$, should be trusted more than those with low or no clicks. A limitation of $\hat{P}_{intu}(e|q)$ is that it weighs the foreground and background models in the same way irrespective of the observed foreground clicks. Our final model, $\hat{P}_{intp}(e|q)$ parameterizes the interpolation by the number of observed clicks:

$$\begin{aligned}\hat{P}_{intp}(e|q) = {} & \alpha[w_e(q,e)]\hat{P}_{mle}(e|q) \\ & + (1-\alpha[w_e(q,e)])\hat{P}_{bsim}(e|q)\end{aligned} \quad (3.7)$$

In practice, we bucket the observed click parameter, $w_e(q,e)$, into eleven buckets: {1-click, 2-clicks, ..., 10-clicks, more than 10 clicks}.

Section 5.2 outlines our procedure for learning the model parameters for both $\hat{P}_{intu}(e|q)$ and $\hat{P}_{intp}(e|q)$.

3.4 Summary

Table 1 summarizes the association models presented in this section as well as a strawman that assigns uniform probability to all edges in QEC:

$$\hat{P}_{unif}(e|q) = \frac{1}{\sum_{e' \in E} \phi(q,e')} \quad (3.8)$$

In the following section, we apply these models to the task of extracting product recommendations for general web search queries. A large-scale experimental study is presented in Section 5 supporting the effectiveness of our models.

4 Entity Recommendation

Query recommendations are pervasive in commercial search engines. Many systems extract recommendations by mining temporal query chains from search sessions and clickthrough patterns (Zhang and Nasraoui, 2006). We adopt a similar strategy, except instead of mining query-query associations, we propose to mine query-entity associations, where entities come from an entity database as described in Section 1. Our technical challenge lies in annotating sessions with entities that are relevant to the session.

4.1 Product Entity Domain

Although our model generalizes to any entity domain, we focus now on a product domain. Specifically, our universe of entities, E, consists of the entities in a large commercial product catalog, for which we observe query-click-product clicks, C_e, from the vertical search logs. Our QEC graph is completed by extracting query-click-urls from a search engine's general search logs, C_u. These datasets are described in Section 5.1.

4.2 Recommendation Algorithm

We hypothesize that if an entity is relevant to a query, then it is relevant to all other queries co-occurring in the same session. Key to our method are the models from Section 3.

Step 1 – Query Annotation: For each query q in a session s, we annotate it with a set E_q, consisting of every pair $\{e, \hat{P}(e|q)\}$, where $e \in E$ such that there exists an edge $\{q,e\} \in C_e$ with probability $\hat{P}(e|q)$. Note that E_q will be empty for many queries.

Step 2 – Session Analysis: We build a query-entity frequency co-occurrence matrix, $\mathbf{A}$, consisting of $n_{|Q|}$ rows and $n_{|E|}$ columns, where each row corresponds to a query and each column to an entity.

The value of the cell A_{qe} is the sum over each session s, of the maximum edge weight between any query $q' \in s$ and e[3]:

$$A_{qe} = \sum_{s \in \mathbf{S}} \psi(s, e)$$

where $\mathbf{S}$ consists of all observed search sessions and:

$$\psi(s, e) = \underset{\hat{P}(e|q')}{\arg\max}(\{e, \hat{P}(e|q')\} \in E_{q'}), \forall q' \in s$$

Step 3 – Ranking: We compute ranking scores between each query q and entity e using pointwise mutual information over the frequencies in $\mathbf{A}$, similarly to Eq. 3.2.

The final recommendations for a query q are obtained by returning the top-k entities e according to Step 3. Filters may be applied on: f the frequency A_{qe}; and p the pointwise mutual information ranking score between q and e.

5 Experimental Results

5.1 Datasets

We instantiate our models from Sections 3 and 4 using search query logs and a large catalog of products from a commercial search engine. We form our QEC graphs by first collecting in C_e aggregate query-click-entity counts observed over two years in a commerce vertical search engine. Similarly, C_u is formed by collecting aggregate query-click-url counts observed over six months in a web search engine, where each query must have frequency at least 10. Three final QEC graphs are sampled by taking various snapshots of the above graph as follows: a) TRAIN consists of 50% of the graph; b) TEST consists of 25% of the graph; c) DEV consists of 25% of the graph.

5.2 Association Models

5.2.1 Model Parameters

We tune the α parameters for $\hat{P}_{intu}$ and $\hat{P}_{intp}$ against the DEV QEC graph. There are twelve parameters to be tuned: α for $\hat{P}_{intu}$ and $\alpha(1)$, $\alpha(2)$, ..., $\alpha(10)$, $\alpha(> 10)$ for $\hat{P}_{intp}$, where $\alpha(x)$ is the observed click bucket as described in Section 3.3. For each, we choose the parameter value that minimizes the mean-squared error (MSE) of the DEV set, where model probabilities are computed using the TRAIN QEC graph. Figure 2 illustrates the MSE ranging over [0, 0.05, 0.1, ..., 1].

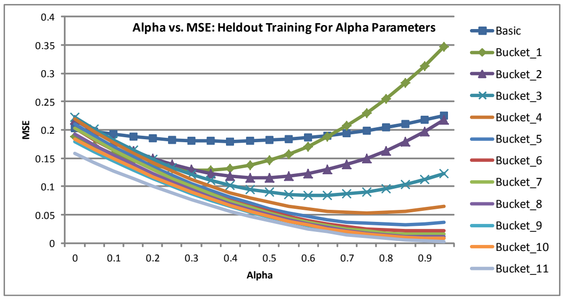

Figure 2: Alpha tuning on held out data.

Model	MSE	Var	Err/MLE	MSE_W	Var	Err/MLE
$\hat{P}_{unif}$	0.0328†	0.0112	-25.7%	0.0663†	0.0211	-71.8%
$\hat{P}_{mle}$	0.0261	0.0111	–	0.0386	0.0141	–
$\hat{P}_{hybr}$	0.0232†	0.0071	11.1%	0.0385	0.0132	0.03%
$\boldsymbol{\hat{P}_{intu}}$	**0.0226†**	**0.0075**	**13.4%**	**0.0369†**	**0.0133**	**4.4%**
$\boldsymbol{\hat{P}_{intp}}$	**0.0213†**	**0.0068**	**18.4%**	**0.0375†**	**0.0131**	**2.8%**

Table 2: Model analysis: MSE and MSE_W with variance and error reduction relative to $\hat{P}_{mle}$. † indicates statistical significance over $\hat{P}_{mle}$ with 95% confidence.

We trained the query synonym model of Section 3.2 on the DEV set and hand-annotated 100 random synonymy pairs according to whether or not the pairs were synonyms [2]. Setting $\rho = 0.4$ results in a precision > 0.9.

5.2.2 Analysis

We evaluate the quality of our models in Table 1 by evaluating their mean-squared error (MSE) against the target $P(e|q)$ computed on the TEST set:

$$MSE(\hat{P}) = \sum_{\{q,e\} \in C_e^T} (P^T(e|q) - \hat{P}(e|q))^2$$

$$MSE_W(\hat{P}) = \sum_{\{q,e\} \in C_e^T} w_e^T(q,e) \cdot (P^T(e|q) - \hat{P}(e|q))^2$$

where C_e^T are the edges in the TEST QEC graph with weight $w_e^T(q, e)$, $P^T(e|q)$ is the target probability computed over the TEST QEC graph, and $\hat{P}$ is one of our models trained on the TRAIN QEC graph. MSE measures against each edge type, which makes it sensitive to the long tail of the click graph. Conversely, MSE_W measures against each edge instance, which makes it a good measure against the head of the click graph. We expect our smoothing models to have much more impact on MSE (i.e., the tail) than on MSE_W since head queries do not suffer from data sparsity.

Table 2 lists the MSE and MSE_W results for each model. We consider $\hat{P}_{unif}$ as a strawman and $\hat{P}_{mle}$ as a strong baseline (i.e., without any graph expansion nor any smoothing against a background

[3] Note that this co-occurrence occurs because q' was annotated with entity e in the same session as q occurred.

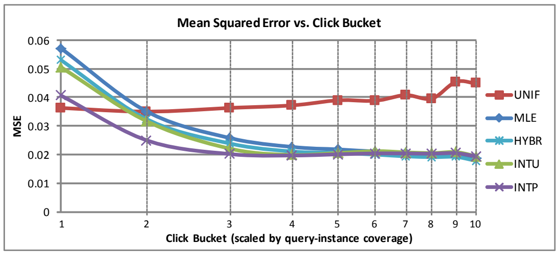

Figure 3: MSE of each model against the number of clicks in the TEST corpus. Buckets scaled by query *instance* coverage of all queries with 10 or fewer clicks.

model). $\hat{P}_{unif}$ performs generally very poorly, however $\hat{P}_{mle}$ is much better, with an expected estimation error of 0.16 accounting for an MSE of 0.0261. As expected, our smoothing models have little improvement on the head-sensitive metric (MSE_W) relative to $\hat{P}_{mle}$. In particular, $\hat{P}_{hybr}$ performs nearly identically to $\hat{P}_{mle}$ on the head. On the tail, all three smoothing models significantly outperform $\hat{P}_{mle}$ with $\hat{P}_{intp}$ reducing the error by 18.4%. Table 3 lists query-product associations for five randomly sampled products along with their model scores from $\hat{P}_{mle}$ with $\hat{P}_{intp}$.

Figure 3 provides an intrinsic view into MSE as a function of the number of observed clicks in the TEST set. As expected, for larger observed click counts (>4), all models perform roughly the same, indicating that smoothing is not necessary. However, for low click counts, which in our dataset accounts for over 20% of the overall click instances, we see a large reduction in MSE with $\hat{P}_{intp}$ outperforming $\hat{P}_{intu}$, which in turn outperforms $\hat{P}_{hybr}$. $\hat{P}_{unif}$ performs very poorly. The reason it does worse as the observed click count rises is that head queries tend to result in more distinct urls with high-variance clicks, which in turn makes a uniform model susceptible to more error.

Figure 3 illustrates that the benefit of the smoothing models is in the tail of the click graph, which supports the larger error reductions seen in MSE in Table 2. For associations only observed once, $\hat{P}_{intp}$ reduces the error by 29% relative to $\hat{P}_{mle}$.

We also performed an editorial evaluation of the query-entity associations obtained with bucket interpolation. We created two samples from the TEST dataset: one randomly sampled by taking click weights into account, and the other sampled uniformly at random. Each set contains results for 100 queries. The former consists of 203 query-product associations, and the latter of 159 associations. The evaluation was done using Amazon Mechanical Turk[4]. We created a Mechanical Turk HIT[5] where we show to the Mechanical Turk workers the query and the actual Web page in a Product search engine. For each query-entity association, we gathered seven labels and considered an association to be correct if five Mechanical Turk workers gave a positive label. An association was considered to be incorrect if at least five workers gave a negative label. Borderline cases where no label got five votes were discarded (14% of items were borderline for the uniform sample; 11% for the weighted sample). To ensure the quality of the results, we introduced 30% of incorrect associations as honeypots. We blocked workers who responded incorrectly on the honeypots so that the precision on honeypots is 1. The result of the evaluation is that the precision of the associations is 0.88 on the weighted sample and 0.90 on the uniform sample.

Query	$\hat{P}_{mle}$	$\hat{P}_{intp}$	Query	$\hat{P}_{mle}$	$\hat{P}_{intp}$
Garmin GTM 20 GPS			Canon PowerShot SX110 IS		
garmin gtm 20	0.44	0.45	canon sx110	0.57	0.57
garmin traffic receiver	0.30	0.27	powershot sx110	0.48	0.48
garmin nuvi 885t	0.02	0.02	powershot sx110 is	0.38	0.36
gtm 20	0	0.33	**powershot sx130 is**	0	0.33
garmin gtm20	0	0.33	**canon power shot sx110**	0	0.20
nuvi 885t	0	0.01	**canon dig camera review**	0	0.10
Samsung PN50A450 50” TV			Devil May Cry: 5th Anniversary Col.		
samsung 50 plasma hdtv	0.75	0.83	devil may cry	0.76	0.78
samsung 50	0.33	0.32	**devilmaycry**	0	1.00
50” hdtv	0.17	0.12	High Island Hammock/Stand Combo		
samsung plasma tv review	0	0.42	high island hammocks	1.00	1.00
50” samsung plasma hdtv	0	0.35	**hammocks and stands**	0	0.10

Table 3: Example query-product association scores for a random sample of five products. Bold queries resulted from the expansion algorithm in Section 3.2.

5.3 Related Product Recommendation

We now present an experimental evaluation of our product recommendation system using the baseline model $\hat{P}_{mle}$ and our best-performing model $\hat{P}_{intp}$. The goals of this evaluation are to (1) determine the quality of our product recommendations; and (2) assess the impact of our association models on the product recommendations.

5.3.1 Experimental Setup

We instantiate our recommendation algorithm from Section 4.2 using session co-occurrence frequencies

[4] `https://www.mturk.com`

[5] HIT stands for Human Intelligence Task

	Query Set Sample				Query Bag Sample			
f	10	25	50	100	10	25	50	100
p	10	10	10	10	10	10	10	10
$\hat{P}_{mle}$ precision	0.89	0.93	0.96	0.96	0.94	0.94	0.93	0.92
$\hat{P}_{intp}$ precision	0.86	0.92	0.96	0.96	0.94	0.94	0.93	0.94
$\hat{P}_{mle}$ coverage	0.007	0.004	0.002	0.001	0.085	0.067	0.052	0.039
$\hat{P}_{intp}$ coverage	0.008	0.005	0.003	0.002	0.094	0.076	0.059	0.045
$R_{intp,mle}$	1.16	1.14	1.13	1.14	1.11	1.13	1.15	1.19

Table 4: Experimental results for product recommendations. All configurations are for $k = 10$.

from a one-month snapshot of user query sessions at a Web search engine, where session boundaries occur when 60 seconds elapse in between user queries. We experiment with the recommendation parameters defined at the end of Section 4.2 as follows: k = 10, f ranging from 10 to 100, and p ranging from 3 to 10.

For each configuration, we report *coverage* as the total number of queries in the output (i.e., the queries for which there is some recommendation) divided by the total number of queries in the log. For our performance metrics, we sampled two sets of queries: (a) `Query Set Sample`: uniform random sample of 100 queries from the *unique* queries in the one-month log; and (b) `Query Bag Sample`: weighted random sample, by query frequency, of 100 queries from the query *instances* in the one-month log. For each sample query, we pooled together and randomly shuffled all recommendations by our algorithm using both $\hat{P}_{mle}$ and $\hat{P}_{intp}$ on each parameter configuration. We then manually annotated each {query, product} pair as *relevant*, *mildly relevant* or *non-relevant*. In total, 1127 pairs were annotated. Interannotator agreement between two judges on this task yielded a Cohen's Kappa (Cohen, 1960) of 0.56. We therefore collapsed the *mildly relevant* and *non-relevant* classes yielding two final classes: *relevant* and *non-relevant*. Cohen's Kappa on this binary classification is 0.71.

Let C_M be the number of relevant (i.e., correct) suggestions recommended by a configuration M and let $|M|$ be the number of recommendations returned by M. Then we define the (micro-) precision of M as: $P_M = \frac{C_M}{C}$. We define relative recall (Pantel et al., 2004) between two configurations M_1 and M_2 as $R_{M_1,M_2} = \frac{P_{M_1} \times |M_1|}{P_{M_2} \times |M_2|}$.

5.3.2 Results

Table 4 summarizes our results for some configurations (others omitted for lack of space). Most remarkable is the $\{f = 10, p = 10\}$ configuration where the $\hat{P}_{intp}$ model *affected 9.4% of all query instances posed by the millions of users of a major search engine*, with a precision of 94%. Although this model covers 0.8% of the unique queries, the fact that it covers many head queries such as *walmart* and *iphone* accounts for the large query instance coverage. Also since there may be many general web queries for which there is no appropriate product in the database, a coverage of 100% is not attainable (nor desirable); in fact the upper bound for the coverage is likely to be much lower.

Query	Product Recommendation
wedding gowns	27 Dresses (Movie Soundtrack)
wedding gowns	Bridal Gowns: The Basics of Designing, [...] (Book)
wedding gowns	Wedding Dress Hankie
wedding gowns	The Perfect Wedding Dress (Magazine)
wedding gowns	Imagine Wedding Designer (Video Game)
low blood pressure	Omron Blood Pressure Monitor
low blood pressure	Healthcare Automatic Blood Pressure Monitor
low blood pressure	Ridgecrest Blood Pressure Formula - 60 Capsules
low blood pressure	Omron Portable Wrist Blood Pressure Monitor
'hello cupcake' cookbook	Giant Cupcake Cast Pan
'hello cupcake' cookbook	Ultimate 3-In-1 Storage Caddy
'hello cupcake' cookbook	13 Cup Cupcakes and More Dessert Stand
'hello cupcake' cookbook	Cupcake Stand Set (Toys)
1 800 flowers	Todd Oldham Party Perfect Bouquet
1 800 flowers	Hugs and Kisses Flower Bouquet with Vase

Table 5: Sample product recommendations.

Turning to the impact of the association models on product recommendations, we note that precision is stable in our $\hat{P}_{intp}$ model relative to our baseline $\hat{P}_{mle}$ model. However, a large lift in relative recall is observed, up to a 19% increase for the $\{f = 100, p = 10\}$ configuration. These results are consistent with those of Section 5.2, which compared the association models independently of the application and showed that $\hat{P}_{intp}$ outperforms $\hat{P}_{mle}$.

Table 5 shows sample product recommendations discovered by our $\hat{P}_{intp}$ model. Manual inspection revealed two main sources of errors. First, *ambiguity* is introduced both by the click model and the graph expansion algorithm of Section 3.2. In many cases, the ambiguity is resolved by user click patterns (i.e., users disambiguate queries through their browsing behavior), but one such error was seen for the query "*shark attack videos*" where several *Shark*-branded vacuum cleaners are recommended. This is because of the ambiguous query "*shark*" that is found in the click logs and in query sessions co-occurring with the query "*shark attack videos*". The second source of errors is caused by *systematic user errors* commonly found in session logs such as a user accidentally submitting a query while typing. An example session is: {"*speedo*", "*speedometer*"} where the intended session was just the second query and the unintended first query is associated with products such as *Speedo swimsuits*. This ultimately causes our system to recommend various swimsuits for the query "*speedometer*".

6 Conclusion

Learning associations between web queries and entities has many possible applications, including query-entity recommendation, personalization by associating entity vectors to users, and direct advertising. Although many techniques have been developed for associating queries to queries or queries to documents, to the best of our knowledge this is the first that aims to associate queries to entities by leveraging click graphs from both general search logs and vertical search logs.

We developed several models for estimating the probability that an entity is relevant given a user query. The sparsity of *query entity graphs* is addressed by first expanding the graph with query synonyms, and then smoothing query-entity click counts over these unseen queries. Our best performing model, which interpolates between a foreground click model and a smoothed background model, significantly reduces testing error when compared against a strong baseline, by 18%. On associations observed only once in our test collection, the modeling error is reduced by 29% over the baseline.

We applied our best performing model to the task of query-entity recommendation, by analyzing session co-occurrences between queries and annotated entities. Experimental analysis shows that our smoothing techniques improve coverage while keeping precision stable, and overall, that our top-performing model affects 9% of general web queries with 94% precision.

References

[Agichtein et al.2006] Eugene Agichtein, Eric Brill, and Susan T. Dumais. 2006. Improving web search ranking by incorporating user behavior information. In *SIGIR*, pages 19–26.

[Agirre et al.2009] Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *NAACL*, pages 19–27.

[Baeza-Yates et al.2004] Ricardo Baeza-Yates, Carlos Hurtado, and Marcelo Mendoza. 2004. Query recommendation using query logs in search engines. In Wolfgang Lindner, Marco Mesiti, Can Türker, Yannis Tzitzikas, and Athena Vakali, editors, *EDBT Workshops*, volume 3268 of *Lecture Notes in Computer Science*, pages 588–596. Springer.

[Baeza-Yates2004] Ricardo Baeza-Yates. 2004. Web usage mining in search engines. In *In Web Mining: Applications and Techniques, Anthony Scime, editor. Idea Group*, pages 307–321.

[Bell et al.2007] R. Bell, Y. Koren, and C. Volinsky. 2007. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *KDD*, pages 95–104.

[Boldi et al.2009] Paolo Boldi, Francesco Bonchi, Carlos Castillo, Debora Donato, and Sebastiano Vigna. 2009. Query suggestions using query-flow graphs. In *WSCD '09: Proceedings of the 2009 workshop on Web Search Click Data*, pages 56–63. ACM.

[Cohen1960] Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, April.

[Craswell and Szummer2007] Nick Craswell and Martin Szummer. 2007. Random walks on the click graph. In *SIGIR*, pages 239–246.

[Fuxman et al.2008] A. Fuxman, P. Tsaparas, K. Achan, and R. Agrawal. 2008. Using the wisdom of the crowds for keyword generation. In *WWW*, pages 61–70.

[Gao et al.2009] Jianfeng Gao, Wei Yuan, Xiao Li, Kefeng Deng, and Jian-Yun Nie. 2009. Smoothing clickthrough data for web search ranking. In *SIGIR*, pages 355–362.

[Good1953] Irving John Good. 1953. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(3 and 4):237–264.

[Jagabathula et al.2011] S. Jagabathula, N. Mishra, and S. Gollapudi. 2011. Shopping for products you don't know you need. In *To appear at WSDM*.

[Jain and Pantel2009] Alpa Jain and Patrick Pantel. 2009. Identifying comparable entities on the web. In *CIKM*, pages 1661–1664.

[Jelinek and Mercer1980] Frederick Jelinek and Robert L. Mercer. 1980. Interpolated estimation of markov source parameters from sparse data. In *In Proceedings of the Workshop on Pattern Recognition in Practice*, pages 381–397.

[Katz1987] Slava M. Katz. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. In *IEEE Transactions on*

Acoustics, Speech and Signal Processing, pages 400–401.

[Kneser and Ney1995] Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 181–184.

[Kurland and Lee2004] O. Kurland and L. Lee. 2004. Corpus structure, language models, and ad-hoc information retrieval. In *SIGIR*, pages 194–201.

[Lidstone1920] George James Lidstone. 1920. Note on the general case of the bayes-laplace formula for inductive or a posteriori probabilities. *Transactions of the Faculty of Actuaries*, 8:182–192.

[Linden et al.2003] G. Linden, B. Smith, and J. York. 2003. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80.

[Liu and Croft2004] X. Liu and W. Croft. 2004. Cluster-based retrieval using language models. In *SIGIR*, pages 186–193.

[Mei et al.2008a] Q. Mei, D. Zhang, and C. Zhai. 2008a. A general optimization framework for smoothing language models on graph structures. In *SIGIR*, pages 611–618.

[Mei et al.2008b] Q. Mei, D. Zhou, and Church K. 2008b. Query suggestion using hitting time. In *CIKM*, pages 469–478.

[Nie et al.2007] Z. Nie, J. Wen, and W. Ma. 2007. Object-level vertical search. In *Conference on Innovative Data Systems Research (CIDR)*, pages 235–246.

[Pantel and Lin2002] Patrick Pantel and Dekang Lin. 2002. Discovering word senses from text. In *SIGKDD*, pages 613–619, Edmonton, Canada.

[Pantel et al.2004] Patrick Pantel, Deepak Ravichandran, and Eduard Hovy. 2004. Towards terascale knowledge acquisition. In *COLING*, pages 771–777.

[Pantel et al.2009] Patrick Pantel, Eric Crestan, Arkady Borkovsky, Ana-Maria Popescu, and Vishnu Vyas. 2009. Web-scale distributional similarity and entity set expansion. In *EMNLP*, pages 938–947.

[Paşca and Durme2008] Marius Paşca and Benjamin Van Durme. 2008. Weakly-supervised acquisition of open-domain classes and class attributes from web documents and query logs. In *ACL*, pages 19–27.

[Ponte and Croft1998] J. Ponte and B. Croft. 1998. A language modeling approach to information retrieval. In *SIGIR*, pages 275–281.

[Sarwar et al.2001] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. 2001. Item-based collaborative filtering recommendation system. In *WWW*, pages 285–295.

[Tao et al.2006] T. Tao, X. Wang, Q. Mei, and C. Zhai. 2006. Language model information retrieval with document expansion. In *HLT/NAACL*, pages 407–414.

[Wen et al.2001] Ji-Rong Wen, Jian-Yun Nie, and HongJiang Zhang. 2001. Clustering user queries of a search engine. In *WWW*, pages 162–168.

[Witten and Bell1991] I.H. Witten and T.C. Bell. 1991. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, 37(4).

[Zhai and Lafferty2001] C. Zhai and J. Lafferty. 2001. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR*, pages 334–342.

[Zhang and Nasraoui2006] Z. Zhang and O. Nasraoui. 2006. Mining search engine query logs for query recommendation. In *WWW*, pages 1039–1040.