Mapping User Search Queries to Product Categories

Carolyn T. Hafernik[†], Bin Cheng[‡], Paul Francis[‡], and Bernard J. Jansen[†]
[†]College of Information Sciences and Technology, The Pennsylvania State University, USA
[‡]Max Planck Institute for Software Systems, Germany
cth132@ist.psu.edu, {bcheng, francis}@mpi-sws.org, jjansen@ist.psu.edu

ABSTRACT
Gathering detailed information about user product interests is becoming increasingly important for online advertisers. However, when gathering this information, maintaining the privacy of online users is a concern. This research is part of a larger project aiming to provide privacy preserving advertising. Specifically, this research aims to provide a method for mapping user search queries to actual product categories while preserving the users' privacy by storing user information on the local host. Product information gathered from two large shopping websites and real user search queries from a log file are used to match user search queries with the most relevant product categories. In matching search queries to product categories, we explore several issues including the algorithm used to rank product categories, the index size, which fields in the index are searched (product description or product name), and what type of product categories are used. Our findings indicate that the most successful algorithms on the user's computer, which preserve privacy, can match the results of those where information is sent to a central server. In addition, the description field of products is the most useful, particularly when searched as a phrase. Having specific fine-grained product categories would help advertisers, search engines and marketers by providing them more information about users while preserving user privacy.

Keywords
Web Search, Online Shopping, Product Search, Privacy Preserving

INTRODUCTION

Obtaining information about users is a central issue in advertising as well as in information search and retrieval research today. A growing area of research is in providing targeted or personalized search results and ads. Information about users is crucial because with more user information, a search engine or ad network can provide results that are personalized for that user as opposed to results that are targeted for the typical user. Additionally, information about users aids in disambiguating queries. For instance, with information about a user's geospatial location, the search results or ads can be targeted for that location (e.g. a search for restaurants can return only restaurants in the user's current location). Employing user information to target search results or ads towards specific users is a form of personalization, in which the content of webpages (e.g. ads or search results) is modified, based on user characteristics such as interests, past behavior, preferences etc. Personalization is increasingly being used as a method of improving search results (Ferragina & Gulli, 2008; Khopkar, Spink, Giles, Shah, & Debnath, 2003; Sieg, Mobasher, & Burk, 2007; Teevan, Dumais, & Horvitz, 2005).

Gathering detailed information about user product interests is becoming increasingly important for online advertisers. However, when gathering this information, maintaining the privacy of online users is a concern. This research is part of a larger project aiming to provide privacy preserving advertising (Privad) (Guha, Cheng, & Francis, 2011; Guha, Reznichenko, Tang, Haddai, & Francis, 2009). We explore how to gather user information while preserving user privacy by limiting how much user information leaves a user's computer. This information could then be used for personalization while characteristics of the user are protected. This approach necessitates moving some of the computational aspects of online advertising to the user's computer.

In order to determine whether this is a viable approach, we aim to take user search terms and queries and map them to product categories collected from a detailed product catalog from a large online shopping site. This mapping provides detailed information about the product categories a user is interested in, aiding in the iden-

ASIST 2011, October 9-13, 2011, New Orleans, LA, USA.

tification of results/selection of ads relevant to the specific interests of the user.

There are several challenges to mapping search terms to product categories. One challenge is *differentiating between product and non-product queries*. If a query is not product related it should not be mapped to a product category. For past research (Jansen, Booth, & Spink, 2008; Rose & Levinson, 2004) focusing on broader categories, this was not an issue as the categories are not product specific; however, our focus is specifically on product related queries in the ecommerce domain. A second challenge is *detecting the relevant categories for a given query*. Is a query related to one category, two categories, five categories? Does the number of categories depend on the query? A third challenge is *ranking the product categories for any given query*. Ideally, we will compare the relevance of categories for different queries, allowing us to keep track of a user's interests over time and to compare the level of interest a user shows in different product categories. We address all three of these challenges in our methodology.

Our work has important implications for researchers, search engines, marketers, businesses and searchers both in presenting relevant ads to users and preserving user privacy. Marketers and businesses can potentially use our research for providing more relevant ads to users. Searchers will benefit from our research as it will help to provide them with more relevant information for searching for products while preserving their privacy.

The remainder of the paper is organized as follows: first we briefly discuss related work and give some necessary definitions that set the stage for our research. Next, we present our research questions and goals. Third, we describe the methodology used in our research. Fourth, we describe our results before discussing them and what issues to consider. Last we offer some final conclusions on the value of this research.

RELATED WORK AND DEFINITIONS

Related Work

There are several areas of related work associated with our research, including personalization, query classification, and product classification. Research on personalization of search has typically focused on re-ranking results and modifying which results are shown to a user (Ferragina & Gulli, 2008; Khopkar et al., 2003; Sieg et al., 2007; Teevan et al., 2005). In addition to research explicitly on personalization, other research has explored gathering information that can be used as a building block for personalization. This work includes classifying queries into broad categories such as user intent either manually (Broder, 2002; Rose & Levinson, 2004) or automatically (Lee, Liu, & Cho, 2005; Kang & Kim, 2003; Baeza-Yates, Caldeŕon-Benavides, & Gonźalez, 2006; Jansen et al., 2008). Our classification of queries is done automatically and on the user's computer.

In addition to work done on general query classification and user intent, research has been done specifically on product query classification (Shen, Li, Li, & Zhou, 2009). The stages of the buying funnel are another way of classifying product queries (Lutze, 2009; Nimetz, 2007; Ramos & Cota, 2008). The KDD Cup 2005 (Kardkovacs, Tikk, & Bansaghi, 2005; Li, Zheng, & Dai, 2005; Shen, Sun, Yang, & Chen, 2006) focused on classifying queries to predefined categories. Many of the categories for this task are related to products. However, the categories are still fairly general, spanning all types of internet searches.

By classifying queries into categories, a search engine gains useful information for implementing personalization. However, broad categories may not give the search engine enough information about the user. A broad category such as shopping does not provide as much information about what the user is looking for as a specific shopping category (e.g. digital cameras) might. Having more precise fine-grained categories or specific product-related categories would give search engines and marketers more information about the user, which would in turn increase the likelihood that the information will be helpful for personalization.

Thus, although classifying queries into broad classes has been examined, little research has examined classifying searches into fine-grained categories or specific product related categories on a user's local computer. While researchers have primarily focused on classifying queries into broad general categories, shopping websites use narrower categories. Shopping websites can have anywhere from a couple of narrow categories to hundreds or tens of thousands of specific product categories, depending on the site. However, any classification done on shopping websites is centralized, proprietary and not private. We focus on classifying queries into narrow categories such as those used by shopping websites while still preserving users' privacy.

De nitions

There are several terms that need to be defined.

User Interest: What do we mean by user interest? Users can have a variety of interests such as product interests, political interests, religious interests, historical interests, research interests, and travel interests. For our research, we focus on product interests. We are interested in what types of products a user might be looking for in preparation for an ecommerce transaction.

Product: The focus on product interests brings us to our second important definition: what do we mean by

products? Like user interests there are many different types of products. One common way of separating products is to separate them into material products and service products. Table 1 gives the two types of products with definitions and examples: goods/products and services (Lovelock, 1983; Murphy & Enis, 1986).

While, in the future, we would like to include all types of products that ads could be geared towards, we currently focus on material products or items with a material form. It should be noted that what counts as material has changed overtime. For example, historically one could argue that a material product had to be something that could be mailed to someone, but today people can also download things. For example, in the past a music CD would be considered a material product as one would have had to purchase an actual physical CD. Today one could download the music from places such as itunes or Amazon.com.

Type	Definitions	Examples
Goods	Products that are actual objects	Book, CD, Car, Clothes, TV
Services	Non-material products	Lawyer, Haircut, Restaurant, Flight tickets

Table 1: Two types of products with examples

RESEARCH QUESTIONS AND GOALS

We attempt to answer the following two questions in this paper.

1. How to separate product and non-product queries from each other?

This is an important issue because we want to match product related queries to product categories. This is necessary for doing classification of queries privately on a user's computer. Additionally, we want to identify product related queries without having to send information off the computer and thus preserve the user's privacy. It is important to recognize which queries can potentially be mapped to product categories and which cannot.

2. How to map user queries to relevant product categories?

This is the main goal of our research: to take a potentially product related query and map it to a relevant product category while storing user information on one client computer. This will be the main focus of this paper. We aim to map user queries to product categories collected from detailed product catalogs of several large online shopping sites. This mapping provides detailed information about the product categories a user is interested in, aiding in the identification of results relevant to the specific interests of the user. By classifying queries into categories, a search engine or an advertiser gains useful information for implementing personalization. While exploring this question we examined the effect of several different variables on our attempts to match queries to product categories:

1. The ranking algorithm used to choose the top category
2. The type of search done over the index (e.g. what fields from products are useful for matching queries to categories)
3. The index size
4. The usage of different product catalogs to build different indexes

We aim to answer both of these questions using the simplest methods possible while preserving user privacy. The ultimate goal is that additional information provided by knowing product categories a user might be interested in can be leveraged to provide more relevant and targeted advertisements to users while still preserving user privacy.

METHODOLOGY

Our basic methodology as shown in Figure 1 is to take a query and classify it into a product category based on the results returned by searches on an inverted index. The index stores information on products that can be used to identify queries with product categories.

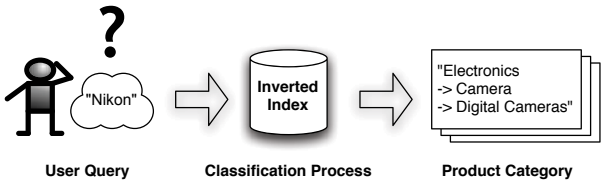

Figure 1: Diagram of the classification process

There are two parts to the following discussion of our methodology. We discuss the different data sets (queries, product catalogs, and product category trees) and then the experimental setup, including the baseline, the index and the ranking algorithms.

Data Sets

The set of queries we used were taken from a large transaction log containing daily information from approximately 3.5 million searches performed by 65,000 users. The log was from the AOL Search Service from March to May 2006. Each record in the log contained

Type	Definitions	Examples	# of queries
Non-product	Not related to products	bad credit personal loans	108
Ambiguous	May be related to actual products so it is useful to match them to a product category	frank sinatra	101
Product	Related to actual products and thus can be matched to a product category	prom dresses	136

Table 2: Three types of queries with examples

an anonymous user id, the date and time the query was submitted, and the click URL. We selected 345 unique queries from the log to test our methods on. Unique queries means that each query was different from the others in our dataset. These queries were selected from the most common queries in the overall dataset. This was to ensure that we used actual user queries for testing our methods. The queries were analyzed and manually separated into three groups: Ambiguous queries, Non-Product queries and Product queries (Table 2).

Dataset from Shopping.com

From Shopping.com, we gathered data using the online API of Shopping.com (Shopping.com, 2010). All of the data gathered from 289 product categories of Shopping.com were used for our index. The products from Shopping.com are placed into a product category tree. Each product has one category associated with it. The categories have multiple hierarchical levels ranging from 1 to 4. Figure 2 shows an example of a leaf category from the Shopping.com product category tree.

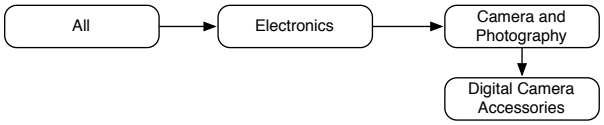

Figure 2: Category sample from Shopping.com.

Dataset from Amazon.com

For Amazon.com, product data were gathered via an online API (Amazon.com, 2010). In total, ∼400-500 GB of product data were gathered from over 100,000 categories. This represents only a small fraction of the amount of product data Amazon.com has available. For Amazon.com, we randomly selected 1,000 products from each category and used those to build the index. We used ∼86 million products from Amazon.com.

The category tree for Amazon.com is larger (thousands of categories as opposed to hundreds) and can be more specific than the one for Shopping.com. (Figure 3 shows an example category from Amazon.com.) In addition, it is not a true tree. There are multiple paths to any given leaf category. Furthermore, any node in the tree can be a category. For instance, using the example from Figure 3, the entire category would be "Electronics → Categories → Camera & Photo → Point & Shoot → Digital Cameras". However, another category could be obtained by cutting off lower level category nodes (e.g. "Electronics → Categories → Camera & Photo") Similarly, within Amazon.com a product can be listed under multiple categories, which makes identifying only one category more difficult.

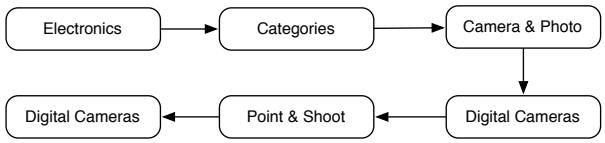

Figure 3: Category sample from Amazon.com.

Experimental Setup

There are two parts to our experiment: First, we established a baseline by sending queries to the shopping site. Second, we designed a method of using an inverted index on the user's computer to categorize the queries. Our method (shown in Figure 1) consists of taking a user query, pre-processing it, submitting it to an inverted index, processing the results of the index, and finally returning one product category.

Baseline Setup

For a baseline, we submitted all of the queries to the Shopping.com API and Amazon.com API. We considered several different ranking schemes for the baseline such as number of products returned for a category and percent of the products in a category returned. The best results were obtained by ranking the results based on the number of products returned from a category. Thus this is the ranking algorithm used for the baseline. These ranked results are compared to those of the indexes we built to provide a baseline.

Index Creation

We created our index using the tool Apache Lucene (ApacheLucene, 2010a), a tool for building and searching indexes of text. To create the index, we first created documents to place in the index. For Shopping.com, we tried two different types of documents: one document for each product and one document for each product category. For Amazon.com, we used only one document per product. Within the product data, each product has multiple types of information (e.g. product name, product description, customer reviews, keywords, price,

seller id, etc.) Amanzon.com and Shopping.com products have similar types of fields, but the names of the fields and how they are accessed vary.

For our purposes, we chose three important fields to build the index: the product name, the product description and the product category. The name and description fields were used to search the index. We chose these fields because they were necessary (i.e., product category) or because they were consistently present in most products and contained information that sellers provided about products. On the other hand, we did not include fields such as customer reviews because they may not be related to the product or may have little information that can be used to identify a product category. Also, fields like customer reviews are not present for all products.

The three indexes created were different sizes (the Shopping.com ones were ∼3 GB and the Amazon.com one was ∼90 GB). This allowed us to explore whether a small index was adequate for our purposes.

Index Search

Before queries were submitted to the index, they were preprocessed. During the preprocessing, stop words were removed, numbers were removed and the individual terms were stemmed using a porter stemmer. Our index can be searched in four different ways (Table 3). We compared the results of the different searches to see which fields where most helpful for identifying relevant product categories.

Type	Abbr.	Definition
Description Search	D	Search product descriptions only
Description phrase search	DP	Search product descriptions using an exact phrase
Name search	N	Search product names only
Name phrase search	NP	Search product names using an exact phrase

Table 3: Four types of searches of the index

Ranking Algorithms

We tried several different ranking algorithms for choosing what the most relevant product category was for any given query. First, we tried Apache Lucene's default weighting scheme. This combines two information retrieval models, the Boolean model and the vector space model, and is based on term frequency (tf) and inverse document frequency (idf). (For the full details of this weighting scheme, see ApacheLucene (2010b).) The second method of ranking was to simply rank categories by the number of products returned from that category. This is the most similar to the ranking of the baseline results. Ranking the product categories by number of products was done on each individual type of search (e.g. description search, description phrase, name phrase and name searches). This led us to our last algorithms where we combined the results of different searches on the index (e.g. description and name searches) to take advantage of the best of each. Table 4 shows the nine methods of combining results. The algorithms vary based on how many of the product categories returned were considered (e.g. the top ten from each search or all) and how we chose the top category from the results of the different searches. Each algorithm used the list of product categories for a given query and returned the resulting top product category. In order to discover algorithms that would be useful for combining the different searches, we looked at the patterns within the results for each type of search.

Name	Procedure
A	1. Sum up number of products for each category 2. Sort by sum of products
B	1. Sort results by # of searches 2. Sort by # of products
C	1. Get top 10 results from each search 2. Do algorithm B
D	1. Get top 10 results from each search 2. Sort by # of products
E	1. Get top 10 results from each search 2. Sort by minimum rank by # of products 3. Sort by count of products 4. Sort by Max Rank 5. Sort by # of Products
F	1. Get top 10 results from each search 2. Sort by average rank by # of products 3. Sort by count returned 4. Sort by # of products
G	1. Do algorithm E with all results
H	1. Do algorithm F with all results
I	1. If top DP result = top NP result then Return DP 2. If top DP result = top D result then Return DP 3. If top D result = top NP result then Return D 4. If top D result = top N result then Return D 5. If top NP result = top N result then Return NP 6. If top DP result = top N result then Return DP 7. If have DP result then Return DP 8. If have NP result then Return NP 9. If have N result then Return N 10. If have D result then Return D

Table 4: Algorithms for combining results of different searches

Evaluating Results

In order to evaluate our results, we took the top ranked product category returned for a query and manually evaluated whether the category was actually relevant. The results of the baseline, the indexes and the various ranking algorithms were then compared to each other. Chi-Square tests were used to see whether differences

between different ranking algorithms/search types were statistically significant.

RESULTS

Identifying Product Searches

The first step we took for classifying queries into product categories was identifying which queries might have relevant product categories. As not all queries have product categories, it is necessary to identify those that do in order to classify queries into product categories privately on the user's computer. For instance, the query "bad credit personal loans" would most likely not be related to any type of material product and thus we do not want to map it to a product category. Here we attempted to automatically separate out queries that had been manually classified as non-product. We had hoped that those queries that were not product related could be easily separated from the product related ones based on the results returned from the indexes. Unfortunately, this was not the case. We examined whether using only the terms in a query and information about the product categories a query was mapped to was helpful for separating product and non-product queries. Unfortunately, this method resulted in many false positives (queries that were not product related being identified as product related). As shown in Figures 4 and 5, on average, there are differences between our three query groups ambiguous, product and non-product. Unfortunately, the differences have no practical significance.

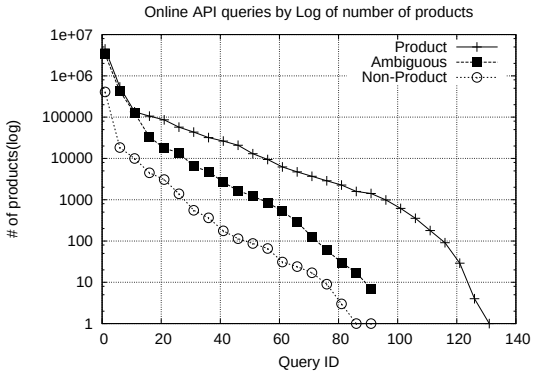

Figure 4: Result based on online API queries

Thus, we need a solution for identifying product related queries on the user's computer that uses information other than search terms and product categories. The solution we developed is two fold. First, we install a list of known shopping websites on the user's computer, and then classify queries made on those websites. The list of shopping sites is taken from websites that have offers (e.g. they are selling products) on Shopping.com or through Google Products. Between the two sites

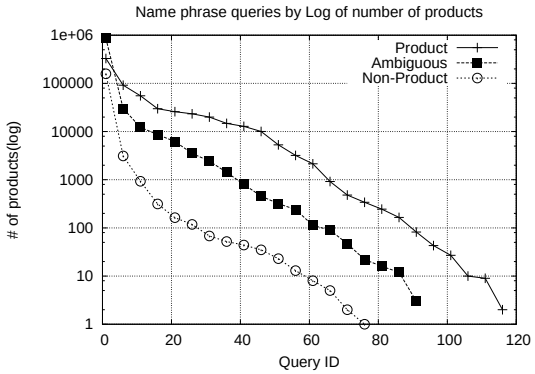

Figure 5: Result based on name phrase queries

there are approximately 18,000 unique domain names for shopping websites. Second, we classify queries for which Google search shows Google Products in the first page of results. We did a brief measurement study of how accurate Google is at identifying a product related query given a general search query. We used the same queries as for other portions of this research. We evaluated the accuracy by examining whether a query returned GoogleProducts / GoogleShopping results in the first page of results. We found that Google has few false positives, meaning that most of the queries it identified as being product related actually were product related. Unfortunately, while there were few false positives there were many false negatives. (Approximately 2/3 of the product queries were not correctly identified as such.) Thus, many user product searches will not be classified as such.

Mapping Queries to Product Categories

All 345 queries were submitted to the index to be mapped to product categories. The results presented below and discussed in the discussion section focus on the mapping of those queries classified as product or ambiguous to product categories.

Shopping.com

Table 5 shows the results for the one product per document index and the Shopping.com API results. Using one product per document results in significantly better results for matching queries to product categories than the second index based on one category per document. In addition, the default weighting scheme from Lucene does not provide good results in our case. This is most likely because the text being searched is short as opposed to longer documents that have multiple paragraphs. The simplest ranking algorithm, that of ranking categories by the number of products returned, provides the best results. In addition, as seen in Table 6, DP was the only individual search type to match the

Field	Ranked by	Ambiguous			Product			Not Product
		relevant	total	%	relevant	total	%	total
Description	# Products	65	101	64	102	136	75	108
	Weight	18	101	18	23	136	17	108
Description phrase	# Products	84	98	86	103	118	87	92
	Weight	18	98	18	25	118	21	92
Name	# Products	60	100	60	99	135	73	107
	Weight	25	100	25	21	135	16	107
Name Phrase	# Products	74	95	78	92	119	77	77
	Weight	25	95	26	25	119	21	77
Combined	Algorithm A	63	101	62	101	136	74	108
	Algorithm B	75	101	74	107	136	79	108
	Algorithm C	73	101	72	107	136	79	108
	Algorithm D	63	101	62	101	136	74	108
	Algorithm E	78	101	77	113	136	83	108
	Algorithm F	66	101	65	104	136	76	108
	Algorithm G	80	101	79	111	136	82	108
	Algorithm H	62	101	61	105	136	77	108
	Algorithm I	81	101	80	116	136	85	110
Online	Online API	74	95	78	118	132	89	93

Table 5: Results for Shopping.com in the case of one document per product

Type	**D**	**DP**	**N**	**NP**	**Algorithm I**	Online API
D		4.46*	0.43	0.1	10.65*	7.17*
DP	4.46*		7.6*	4.89*	1.37	0.33
N	0.43	7.6*		0.31	16.29*	11*
NP	0.1	4.89*	0.31		11.31*	7.72*
Algorithm I	10.65*	1.37	16.29*	11.31*		0.36
Online API	7.17*	0.33	11*	7.72*	0.36	

Table 6: Results of Chi-Square Test for Shopping.com data (* indicates a statistical difference at the p=0.05 level)

Online API. It also gave a statistically significant improvement over D, N and NP. For the combined search types, the different algorithms ranged between matching the best results of the D or N searches and matching the best results of the DP and NP results. Of the combined algorithms, algorithm I produced the best results matching both the Online API and DP. This indicates that we can achieve similar results without revealing private information of the user.

Amazon.com

Amazon.com shows similar results to Shopping.com. As with Shopping.com, the Lucene weighting scheme did not work very well. Table 7 shows the results for ranking the results by number of products. Initially the results were significantly lower than the API. This is partially explained by the high number of queries with a "book" category as the top category. Few of the queries were actually for books. We removed the categories starting with "book." As seen in Table 8, this improved results by ∼10% or more, bringing the results closer to the Amazon.com API results.

Comparison

Figure 6 shows a comparison between Shopping.com and Amazon.com indexes and the APIs. As can be seen, Shopping.com on average does better than Amazon.com. Interestingly a brief examination of which queries did better for which index shows that while Shopping.com does better in general, Amazon.com does better for very specific queries. For instance, if one searches for a specific type of toy, then matching the query to an Amazon.com category can be very accurate as there may be a category for just that type of toy. On the other hand, if a user's query shows a general interest in toys but not in any specific toys, then the query can be more accurately mapped to a category from shopping.com. We had more trouble matching queries to more general Amazon.com categories than we did matching them to more specific categories.

DISCUSSION

Results

We had two research questions: First, how to separate product and non-product queries from each other? Sec-

Type	Relevant	# of Queries	%
Ambiguous			
description	37	100	37
description phrase	57	99	58
name	45	98	46
name phrase	51	92	55
online API	72	101	71
Product			
description	50	135	37
description phrase	73	122	60
name	74	134	55
name phrase	80	114	70
online API	117	132	89

Table 7: Amazon.com results ranked by number of products

Type of Search	Relevant	# of Queries	%
description	84	133	63
description phrase	87	118	74
name	93	133	70
name phrase	86	110	78

Table 8: Amazon.com results for product queries without books ranked by number of products

ond, how to map user queries to relevant product categories?

For the first research question, identifying product queries, we discovered that we could separate queries out into statistically significant groups. Unfortunately, although the differences between the types of queries were statistically significant, they were not practically significant.

For the second research question, matching queries to product categories, we explored several issues related to the index. First, how should one go about ranking the results returned by the index. As described in the methodology and result sections, we attempted several different ranking algorithms ranging from the default Apache Lucene weighting scheme, to the simple rank by number of products algorithm, to more complicated ways of combining multiple types of searches. Our results show that for both Shopping.com and Amazon.com ranking the results by the number of products was the most successful when compared to the online APIs. Additionally, while combining the multiple searches did not necessarily hurt results, this did not provide significantly better results.

A connected issue was which product fields are actually useful for matching queries to product categories. Any given product has many fields (e.g. product name,

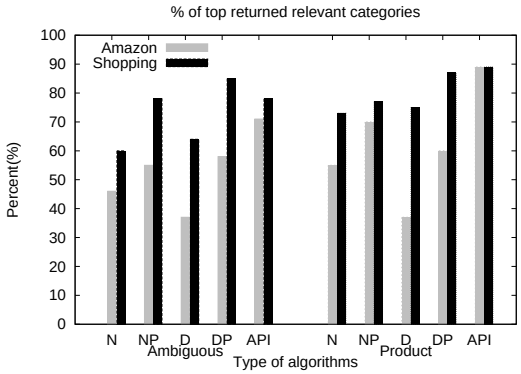

Figure 6: Results comparison between Shopping.com and Amazon.com

product description, customer reviews, product price etc.). We chose to use only two of these fields: product name and product description. These were two fields that almost all products had and that appeared most relevant. Still, we wanted to see which would be better. For Shopping.com it is clear that searching the product descriptions is more useful than searching the product names. However, for Amazon.com which field is most useful is inconclusive in that neither of the fields did significantly better than the other. Thus, more exploration of which fields are most useful is needed because currently which one does better varies based on whether Amazon.com or Shopping.com data are used for the index and on whether phrases are used.

Another consideration we explored was index size. Ideally, the index used should be small so that users can quickly download it onto their computers. Thus, a 10 GB index is too large to reasonably work for our system. The index needs to be downloaded quickly and either easily replaced or updated. To determine whether a small index was adequate, we built two indexes of different sizes, one ∼3GB and the other ∼90GB. The smaller index was based on products from Shopping.com while the larger index was based on products from Amazon.com. On average the small index, based on Shopping.com products worked better for us. This is promising as it shows that one does not need a huge index to attain reasonable results. However, the size of the index is still too large to be quickly downloaded. Shrinking the index further would be useful, but so far experiments with doing this have proved unsuccessful.

What type of product categories would be most useful? Are very general categories such as "clothes" more useful or are more specific categories such as "clothes → women's → skirts" more useful? To look at this, we used two different product category trees, one of

which is very general and one of which is more specific. (See the methodology section for a full discussion of these two product category trees and their similarities and differences.) As can be seen from our results and from Figure 6, on average, we were more successful at matching queries to more general product categories (as in the Shopping.com categories) than to more specific product categories. However, it would be interesting to explore mapping very specific product related queries to categories. The fact that most of our queries were shorter and more general may have affected which types of product categories were better matches.

A last issue is how to evaluate relevance. This includes several sub issues. First, any given query could be matched to multiple categories that are equally relevant. For instance, the query "star wars" could map to categories including "toys and games," "books," "music," and "movies". All of these categories are in some sense relevant. Without a more specific and less ambiguous query it would be hard to match a query to only one category. On the other hand, some queries might have only one specific category that would be relevant. It depends on how specific and unambiguous the query is. For our purposes, we do not need an absolute list of all of the categories that a query is relevant for. We only need a reasonable idea of what type of product would be relevant for a query. Thus, we only examined the top ranked product category and its relevance to the query. If it is relevant, then the query matching was successful. While it would be better to have multiple relevant product categories for each query, it is still useful to have only the top one. In addition, on average as the rank of the product category goes down it becomes more and more unlikely that the category will actually be relevant.

Even assuming one only takes the top ranked product category, the issue still remains of how does one tell if this category is relevant or not. For us, this meant manually looking at the query and the top category and examining whether they appeared to be connected. However, this is obviously a subjective process, especially with only brief queries, which are naturally ambiguous to start with. Nevertheless, some queries are much easier to evaluate than others. Matching the query "prom dress" with the category "clothes → women's clothes → dresses" is a clear match. However, other queries are not as easily matched.

Practical Implications

Our research has implications for multiple groups, especially online advertisers and people concerned with privacy. Our method shows that it is possible to gather detailed information about a user without revealing private information of the user. We preserve privacy by storing the index on the user's computer and only sending a product category off of the computer. This means that we do not need to send the exact search terms off of the computer allowing us to preserve the user's privacy. This detailed information can be used by advertisers and companies to provide better targeted ads. This could be useful for search engines and other ad providers as they can provide better targeted ads while assuring users that their privacy is being preserved. Our specific goal for this research was to provide information that could be used to supply or choose ads displayed to users. However, there is no reason why similar techniques could not be used to provide other types of personalized information to users while preserving their privacy. For instance, the information could also be used to provide better targeted results in general.

Limitations

One limitation is that our queries were from a 2006 transaction log. As product interests and types of available products change frequently, there are likely many queries related to products that were not available in 2006 being used today. However, this is not a major failing, especially since the product data used was collected in 2010. Another limitation is that we only used queries entered on search engines. It is possible that queries entered on an actual shopping site might be different. To this end, we are currently investigating the differences between those types of queries. A third limitation is that we did not have access to actual users and thus had to infer their interests from the terms in the queries. We plan to triangulate this research with user studies and surveys. A final limitation, as discussed earlier, is the size of the index. In today's world, storing a 3GB index would not be a problem for many people, particularly if they have to opt in before downloading it. The main issues with the size of the index would be download speed and how often it needs to be updated. However, updates could be small thus only the initial download would be large.

Strengths

Some of the strengths of our work are as follows. We use actual queries from actual users. Thus, the queries are all ones that a user could potentially use. This increases the validity of our results. We look at results using two different product categorization schemes from different popular shopping websites. We were able to look at two different granularities of categories. Thus our results are more generalizable than if we had used only one product category tree/scheme. In addition, we can comment on the granularity of the shopping categories. We demonstrate that using a fairly small index we can still provide useful information about users. Thus our method would provide detailed information about the product interests of users without private information needing to leave the user's computer.

CONCLUSIONS

The goal of our research was to gather detailed information about user product interests that could then be sent off of users' computers without revealing more detailed private information. Specifically we aimed to take user queries and identify them with a product category in order to aid in providing better-targeted advertising while preserving privacy. Our findings indicate that this is possible as the most successful algorithms on the user's computer, which preserves privacy, can match the results of those where personal information is sent off the computer. Having specific fine-grained product categories would help advertisers, search engines and marketers by providing them with more information about the users. Better information on user product interests would help provide more relevant ads for users. If this information can be provided while still preserving the privacy of the users then users may be more willing to use services that provide personalized ads.

ACKNOWLEDGEMENTS

This work was done while the primary author was an intern at the Max Planck Institute for Software Systems (MPI–SWS).

REFERENCES

Amazon.com. (2010, May). *Amazon.com api.* Retrieved from: `https://affiliate-program.amazon.com/gp/advertising/api/detail/main.html`.

ApacheLucene. (2010a, May). *Apache Lucene.* Retrieved from: `http://lucene.apache.org/java/docs/index.html`.

ApacheLucene. (2010b, May). *Apache Lucene- Scoring.* Retrieved from: `http://lucene.apache.org/java/3_0_3/scoring.html`.

Baeza-Yates, R., Caldeŕon-Benavides, L., & Gonźalez, C. (2006). The intention behind web queries. *Lecture Notes in Computer Science 4209*, 98-109.

Broder, A. (2002, September). A taxonomy of web search. *SIGIR Forum*, *36*, 3–10.

Ferragina, P., & Gulli, A. (2008). A personalized search engine based on web-snippet hierarchical clustering. *Software Practice and Experience*, *38*(2), 189-225.

Guha, S., Cheng, B., & Francis, P. (2011). Privad: Practical privacy in online advertising. *Proceedings of the 8th Symposium on Networked Systems Design and Implementation*.

Guha, S., Reznichenko, A., Tang, K., Haddai, K. H., & Francis, P. (2009). Serving ads from localhost for performance, privacy and profit. *Proceedings of Hot Topics in Networking (HotNets)*.

Jansen, B. J., Booth, D. L., & Spink, A. (2008, May). Determining the informational, navigational, and transactional intent of web queries. *Inf. Process. Manage.*, *44*, 1251–1266.

Kang, I., & Kim, G. (2003). Query type classification for web document retrieval. *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, 64–71.

Kardkovacs, Z. T., Tikk, D., & Bansaghi, Z. (2005, December). The ferrety algorithm for the kdd cup 2005 problem. *ACM SIGKDD Explorations Newsletter*, *7*, 111–116.

Khopkar, Y., Spink, A., Giles, C. L., Shah, P., & Debnath, S. (2003). Search engine personalization: An exploratory study. *First Monday*.

Lee, U., Liu, Z., & Cho, J. (2005). Automatic identification of user goals in web search. *Proceedings of the 14th International Conference on World Wide Web*, 391–400.

Li, Y., Zheng, Z., & Dai, H. K. (2005, December). Kdd cup-2005 report: facing a great challenge. *ACM SIGKDD Explorations Newsletter*, *7*, 91–99.

Lovelock, C. H. (1983). Classifying services to gain strategic marketing insights. *The Journal of Marketing*, *47*(3), 9-20.

Lutze, H. (2009). *The Findability Formula: The Easy, Non-Technical Approach to Search Engine Marketing.* Hoboken, NJ: Wiley.

Murphy, P. E., & Enis, B. M. (1986). Classifying products strategically. *The Journal of Marketing*, *50*(3), 24-42.

Nimetz, J. (2007, March 27). *B2B Marketing in 2007: The Buying Funnel vs. Selling Process.* Retrieved from: `http://www.searchengineguide.com/jody-nimetz/b2b-marketing-i-1.php`.

Ramos, A., & Cota, S. (2008). *Search Engine Marketing.* New York, NY: McGrew-Hill.

Rose, D. E., & Levinson, D. (2004). Understanding user goals in web search. *Proceedings of the 13th International Conference on World Wide Web*, 13–19.

Shen, D., Li, Y., Li, X., & Zhou, D. (2009). Product query classification. *Proceeding of the 18th ACM Conference on Information and Knowledge Management*, 741–750.

Shen, D., Sun, J.-T., Yang, Q., & Chen, Z. (2006). Building bridges for web query classification. *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 131–138.

Shopping.com. (2010, May). *Shopping.com API.* Retrieved from: `http://developer.shopping.com/`.

Sieg, A., Mobasher, B., & Burk, R. (2007). Web search personalization with ontological user profiles. *Proceedings of the 16th ACM Conference on Information and Knowledge Management*, 525–534.

Teevan, J., Dumais, S. T., & Horvitz, E. (2005). Personalizing search via automated analysis of interests and activities. *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 449–456.