Entropy of Search Logs: How Hard is Search? With Personalization? With Backoff?

Qiaozhu Mei [*]
University of Illinois at Urbana Champaign
Urbana,IL 61801
qmei2@uiuc.edu

Kenneth Church
Microsoft Research
Redmond, WA, 98052
church@microsoft.edu

ABSTRACT

How many pages are there on the Web? 5B? 20B? More? Less? Big bets on clusters in the clouds could be wiped out if a small cache of a few million urls could capture much of the value. Language modeling techniques are applied to MSN's search logs to estimate entropy. The perplexity is surprisingly small: millions, not billions.

Entropy is a powerful tool for sizing challenges and opportunities. How hard is search? How hard are query suggestion mechanisms like auto-complete? How much does personalization help? All these difficult questions can be answered by estimation of entropy from search logs.

What is the potential opportunity for personalization? In this paper, we propose a new way to personalize search, personalization with backoff. If we have relevant data for a particular user, we should use it. But if we don't, back off to larger and larger classes of similar users. As a proof of concept, we use the first few bytes of the IP address to define classes. The coefficients of each backoff class are estimated with an EM algorithm. Ideally, classes would be defined by market segments, demographics and surrogate variables such as time and geography.

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]: Text Mining

General Terms: Measurements

Keywords: entropy, search log, search difficulty, personalization with backoff, demographics

1. INTRODUCTION

How many pages are there on the Web? 5B? 20B? More? Less? How hard is search? How much does personalization help? All are difficult but crucial questions to search business.

Scale is hard. The bigger the web, the harder the search. Search engines make large investments in expensive computer centers in the cloud to index billions of pages. Could

[*] This work was done when the first author was on a summer internship at Microsoft Research.

these large investments be wiped out if a small cache of a few million pages could capture much of the value? What if someone found a way to squeeze much of the value of the cluster into a desktop or a mobile device? Is search more like an Everest expedition (clusters in the clouds) or a walk in the park (a little flash memory on a mobile device)?

Related questions come up in language. How big is English? One can find simple answers on the covers of many dictionaries, but we would feel more comfortable with answers from a more authoritative source than a marketing department. Many academics have contributed to this discussion from many perspectives: Education, Psychology, Statistics, Linguistics, and Engineering. Chomsky and Shannon proposed two different ways to think about such questions:

- Chomsky: language is infinite [4]

- Shannon: 1.3 bits per character [24]

These two answers are very different. Chomsky's answer is about the total number of words; and Shannon's answer is about the perplexity, or the difficulty of using a language. A dictionary could cover a lot of words, but not all of them are actively used. Using a Chomskian argument, we could argue that there are infinitely many urls. For example, one could write a spider trap such as successor.aspx?x=0 which links to successor.aspx?x=1 which links to successor.aspx?x=2. In addition to intentionally malicious spider traps, there are perfectly benign examples such as calendars[1], where there are infinitely many pages, one for each month, with links from each month to the next. It is all too easy to build a web crawler that finds itself attempting to materialize an infinite set with finite resources. The crawler can easily consume all available time and space.

Shannon offers a more practical answer. Although there are a lot of pages out there, there are not that many pages that people actually go to. This paper will estimate entropy of urls (and queries and IP addresses) based on logs from Microsoft's www.live.com. We find that it takes just 22 bits to guess the next url (or the next query or the next IP address). That is a walk in the park (millions, not billions). With all the talk about the long tail, one would think that the web was astronomical. But the logs are tiny, far less than Carl Sagan's billions and billions [22].

As we will see, entropy is a powerful tool for sizing challenges and opportunities. How hard is search? How much does personalization help?

[1] http://www.timeanddate.com/calendar/monthly.html?year=2005&month=12&country=11

1.1 Personalization

Personalization is a hot topic, with a large body of work, not only in the scientific literature, but also in commercial practice. Many people use personalized search products every day. A query for "personalized search" returns millions of page hits. The first few pages of results are dominated by the commercial practice. If you want to find the scientific literature such as [29], you'll have to refine the query considerably by adding a keyword like "SIGIR."

Why does personalization help? It is useful to know your audience. Consider the ambiguous query: "MSG". Depending on the user, this query could be looking for the sports arena (Madison Square Garden) or the food additive (Monosodium Glutamate). The search engine could do a better job answering ambiguous queries like this if it had access to demographic data and/or log data such as click logs.

Many acronyms are ambiguous. ACS can refer to the "American Chemical Society," the "American Cancer Society," the "American College of Surgeons" and more. Acronyms take on special meanings inside many large organizations and private enterprises. For example, for most people, MSR means "Mountain Safety Research," but inside Microsoft, it means "Microsoft Research." And of course, it means other things to other people including: "Montessori School of Raleigh," "Mom Service Representative" and "My Sports Radio." PSS is a stock ticker for "Payless Shoes," as well an abbreviation of several different companies: "Physicians Sales and Service," "Phoenix Simulation Software," "Personal Search Syndication," "Professional Sound System," etc. But inside Microsoft, PSS refers to "Product Support Services." It helps to know your audience in order to know:

- what the terminology means

- which questions are likely to come up, and

- which answers are likely to be appreciated.

If we have the relevant data (such as click logs) for a particular user, we should use it.

1.2 Personalization With Backoff

But what if we do not have data for a particular user (or we cannot use it because of privacy concerns)? This paper takes a backoff approach to personalization [17]. If we do not have data for a particular user, back off to larger and larger groups of similar users. As a proof of concept, users are grouped into equivalence classes based on the most significant bytes of their IP address. Personalization is then conducted by combining estimates based on all four bytes of the IP address, the first three bytes, the first two, and so on. It would be even better to group customers by market segments and/or collaborative filtering (users who ask similar questions and click on similar urls). We leave these suggestions for future work.

Segmentation is a traditional goal in marketing. Customers are assigned to equivalence classes based on profile features such as age, income, occupation, etc. It is useful for an advertiser to know who it is talking to so that it can target the message appropriately to the audience. An advertiser such as Ford, for example, has a wide range of products. Some products are more attractive to some customers, and other products are more attractive to other customers. For example, the firm may wish to target small trucks to a rural audience and hybrids to a green audience. Companies

would like to know if they are talking to college students, teenagers, parents with young children, etc. It is useful to know the class of your audience.

We find that a little bit of personalization is better than too much or too little. Specifically, personalization with backoff to higher bytes of IP addresses (especially the second and third bytes) is better than 100% personalization or no personalization. Too little personalization misses the opportunity and too much runs into sparse data (and privacy). It isn't feasible to know everything about everyone (and they might not like it, if we knew too much).

Instead of assigning each customer to his own class (i.e., 100% personalization), it is common to assign customers to market segments. Market segments are typically defined in terms of surrogate variables such as geography (e.g., zip code), and time of day and day of week. These surrogate variables are easy to work with, and hopefully, they are well correlated with the more sensitive demographic variables such as those mentioned above.

2. ENTROPY ESTIMATION

How big is the web? How hard is search? How much does personalization help?

To answer these questions and more, we collected a sample of logs from the Live search engine of about 1.5 year up to July 2007. This 1.5 year data, denoted as the "bigger" dataset, contains 193 million unique IP addresses, 637 million unique queries, and 585 million unique urls. The sample contains about 10 million $\langle Q, URL, IP \rangle$ triples per day. Each triple corresponds to a click from a particular IP address on a particular url for a particular query.

We separated the logs between 1/1/2006 and 2/6/2006 specifically for personalization experiments. The January data (the "smaller" data) was used for training the model of personalization and the February data was used for validation and testing. This one month training set contains 26 million unique IP addresses, 36 million unique queries, and 63 million unique urls. Entropy was estimated based on both the smaller data and the bigger data.

We assume that these sets are reasonably representative of the tasks of interest, though of course, such assumptions can be highly problematic. It is possible, for example, that users could share the same IP address, and a user would click on different urls under different conditions.

2.1 Notation

- **U**: a user

- **IP**: an IP address. IP will be used as a convenient surrogate for U, though of course, it is possible for multiple users to share the same IP address.

- **C**: a class of users. Users are grouped into equivalence classes based on variables such as IP prefixes, time and location. These variables are treated as convenient surrogates for variables of interest such as demographics, market segments, etc.

- **URL**: Uniform Resource Locator, the name of a web document.

- $\langle Q, URL, IP \rangle$: a triple from the search logs, indicating that there was a click on a particular URL in response to a particular query Q from a particular IP address.

2.2 Entropy (H)

Entropy[2] is commonly used in information theory to characterize the size of the search space. The larger the entropy of a distribution is, the harder it is to predict the next event [23]. In [24], Shannon used entropy to measure the difficulty of predicting the next character of English. Shannon's entropy provides bounds, but does not say how to achieve these bounds.

Similarly, we introduce entropy to measure the difficulty faced by a search engine. Note that entropy measures the size of the search space of the web, but not the number of particular urls. This is practical since what a search business cares about is the difficulty of search. How many bits does it take to guess the next url that will be clicked on?

$$H(URL) = -\sum_{URL} p(URL) \log p(URL)$$

Conditional entropy measures the remaining entropy of the target random variable given the value of another related random variable. We can thus use conditional entropy to measure the difficulty of web search, when we know the query, the user, etc. The search task, of course, is much easier, because we are given the query, which is a huge hint.

$$H(URL|Q) = H(URL, Q) - H(Q)$$

How much does personalization help? That is, suppose we give the search engine not only the query, but also the IP address. How much does the IP address help?

$$H(URL|Q, IP) = H(URL, Q, IP) - H(Q, IP)$$

These quantities and more can be estimated from the training data, a sequence of triples: $\langle Q, URL, IP \rangle$. We will present estimates of the entropy of urls, queries and IP addresses, taken one at a time. In addition, we will present estimates of the joint entropies of all pairs of these quantities, as well as the three-way joint. From these quantities, we can easily derive estimates of conditional entropies of any combination of these variables (X) given any other combination of these variables (Y) using the rule[3]

$$H(Y|X) = H(X, Y) - H(X)$$

2.3 Cross Entropy

It is common practice to split the data into two pieces, one for training and the other for validation. Entropy estimation is fundamentally a prediction task. The task is to use historical logs to estimate search experiences in the future. Splitting up the data into separate training and validation sets tend to produce larger (and more credible) estimates of entropy. Cross entropy[4] can be applied to measure the average number of bits needed to guess the next url in new logs (validation), given the distribution estimated from the historical logs (training).

For example, the cross entropy of url given the query and IP address is $H_c(URL|Q, IP)$

$$= -\sum_{URL, IP, Q} p_v(URL, IP, Q) \log p_t(URL|IP, Q)$$

where $p_t(URL|IP, Q)$ is estimated from the training set (historical log data) and $p_v(URL, IP, Q)$ is estimated from the

---

[2]http://en.wikipedia.org/wiki/Information_entropy

[3]http://en.wikipedia.org/wiki/Conditional_entropy

[4]http://en.wikipedia.org/wiki/Cross_entropy

---

validation set (new log data). Please note that minimizing this cross entropy is equivalent to maximizing the likelihood of the new log data, given the estimates from the history.

Based on these evaluation measures, we present a series of experimental results which answers the questions in Section 1. How big is the web? How hard is search? How much does personalization help?

3. HOW LARGE IS THE WEB?

Entropy estimates for Q (query), URL and IP addresses are shown in Table 1. The entropy estimates are surprisingly small; 22 bits is millions, not billions. A cache of a few million pages will cover much of the demand.

Combination	One Month	1.5 Year
H(Q)	21.14 (25.1)	22.94 (29.2)
H(URL)	22.06 (25.9)	22.44 (29.1)
H(IP)	22.09 (24.6)	22.64 (27.5)

Table 1: The search space of the web is surprisingly small; 22 bits of entropy corresponds to a perplexity of millions, not billions.

The numbers in brackets correspond to the entropy if the data is uniformly distributed, or the maximum entropy. The actual estimates are significantly smaller than these upperbounds, and change very slowly with the increase of data.
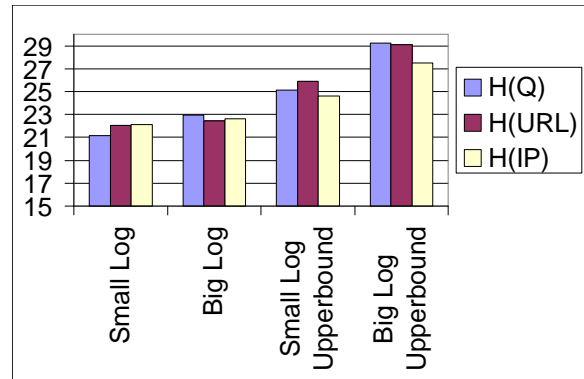
Figure 1: Entropy of logs grows much more slowly than its upperbound

Figure 1 shows a clearer trend. With the bigger dataset, the maximum entropies (upperbounds) increase significantly (> 3 bits, 1 bit corresponds to a twice larger search space). The actual entropies, however, stay around 22 to 23.

3.1 The Population Bound

How large could the web become? Chris Anderson points out in "The Long Tail" (www.thelongtail.com) that online distribution channels are making it possible for NetFlix, Amazon and others to sell less of more [1]. But there are limits to this process. NetFlix offers just 70,000 products[5] and Amazon has just 8 million. That's millions, not billions.

How about vanity searches? Even if everyone looks for their home page, how many pages is that? Right now, there are home pages for famous people and academics, but not ordinary people like our neighbors. There aren't that many

---

[5]http://www.netflix.com/BrowseSelection?lnkctr=nmhbs

famous people and academics: perhaps millions, but certainly not billions.

The telephone business is a mature business that has saturated the market with universal service. Most people (and most businesses) are listed in the white pages and/or yellow pages (unless they opt out). Our neighbors are likely to be in the phonebook, but they don't have a home page on the web.

Phonebooks are limited by the population. According to the FCC,[6] there are about 173 million telephone lines in the US, or less than one per person.

Eventually, when billions of people have universal web service and everyone has their own home page, the web will be bounded by the population, billions of pages worldwide. But, for the foreseeable future (a decade), the web will be a growth market, far from saturation. Millions (not billions) will be good enough until the market saturates.

3.2 Equilibrium: Supply = Demand

In addition to supply side accounting, we can also use a demand side argument to justify the population bound.

Users have only so much time to surf the web. Suppose that each user is willing to spend a few hours per day on the web, assuming that they value the web about as much as telephone (1 hour of usage per day per telephone number)[7] and television (8 hours of usage per day per household)[8]. Assume further that users can only visit so many pages, given these time constraints. Thus, when the web eventually saturates the market, the total number of page hits will be constrained by the size of the population: $O(population)$.

Let's assume there is an equilibrium constraint between suppliers and consumers. Consumers have limited time. They can visit only so many pages within that time limit. Suppliers will compete for these hits. Excluding illegitimate suppliers (spam), reasonable suppliers depend on these hits for their livelihood. Reasonable suppliers will post as many pages as consumers can consume (and no more). Thus, if there is a population bound on hits, then there will also be a population bound on the supply of reasonable pages (that people will look for and click on and value).

3.3 How Hard is Search?

Combination	One Month	1.5 Year
H(Q)	21.14	22.94
H(URL)	22.06	22.44
H(IP)	22.09	22.64
H(Q,URL)	23.88	26.41
H(Q,IP)	26.00	30.41
H(IP,URL)	27.06	31.16
H(Q,URL,IP)	27.17	31.67

Table 2: Entropy estimates of all combinations of Q (query), URL and IP addresses.

Table 2 is like Table 1, but adds joint entropies for all combinations of Q (query), URL and IP address. The size

of the search space for search can be estimated from Table 2. The search task is to guess the URL that the user is looking for from a Query Q. Based on the one month data, that is,

$$H(URL|Q) = H(Q, URL) - H(Q)$$
$$= 23.9 - 21.1 = 2.8$$

This number becomes *3.5* with the bigger data. In other words, search is doable. A user can often find the url he is looking for somewhere in the top 10 search results. That is reassuring, though not surprising.

We would expect an upper bound around $log_2 10 \approx 3.3$ bits, given the source of the data (click logs). Users tend to click somewhere on the first page of results, or not at all.

3.4 How much does Personalization Help?

Suppose we give the search engine not only the query, but also the IP address. How much does that help? Using the one month estimates in Table 2 above,

$$H(URL|Q, IP) = H(Q, URL, IP) - H(Q, IP)$$
$$= 27.2 - 26.0 = 1.2$$

In other words, personalization cuts the search space in half. That is a huge opportunity. This entropy becomes *1.3* with the bigger data. Please note that although the joint entropy of the three variables increases a lot, this conditional entropy remains very small.

Why does personalization help? Consider the ambiguous query: MSG. Some users, especially those near New York City, are looking for the sports arena (Madison Square Garden), whereas other users are looking for the food additive (Mono-sodium Glutamate). The search engine should use the user's history of queries and clicks (when possible) to disambiguate.

3.5 How Hard are Query Suggestions?

There are a number of applications that search the space of queries as opposed to the space of answers. For example, a number of query suggestion mechanisms have been proposed suggest as Google Suggests[9] and The Wild Thing [7]. How hard is it to guess the next question, as opposed to guessing the next answer? $H(Q) = 21.1$ bits (22.9 from 1.5 year).

How much does personalization help?

$$H(Q|IP) = H(Q, IP) - H(IP)$$
$$= 26 - 22 = 4$$

This number becomes *7.8* with the bigger data. In other words, personalization cuts the search space in more than a half. This is a really huge opportunity.

The entropy estimates in this section assume that the search log data is seen, and thus correspond to the lower bounds of search difficulty. In reality, when a search engine tries to predict unseen data, the actual entropies (cross entropies) would be higher. Smoothing methods have to be applied on the personalization language models. We will introduce one possible choice in the following section.

4. PERSONALIZATION WITH BACKOFF

The entropy numbers are really exciting. They make a strong case for plausibility, but there are many remaining challenges that need to be addressed including privacy

---

[6] See Table 7.3 of http://www.fcc.gov/Bureaus/Common_Carrier/Reports/FCC-State_Link/IAD/trend605.pdf.

[7] Table 10.2 of http://www.fcc.gov/Bureaus/Common_Carrier/Reports/FCC-State_Link/IAD/trend803.pdf reports that the average telephone line is used for 71 minutes per day.

[8] http://www.nielsenmedia.com/newsreleases/2005/AvgHoursMinutes92905.pdf

[9] http://labs.google.com/suggests

and data sparsity. In fact, Shannon's entropy gives a lower bound of the search difficulty, but does not provide an operational procedure to achieve it. Personalization is very attractive when we have plenty of data, but what if we do not have enough data, or we cannot use much of the data that we have because of privacy concerns? In this section, we introduce one possible operational procedure of approaching this lower bound: personalization with backoff.

4.1 User Modeling with Backoff

If we don't have enough data for a particular user, or we can't use the data we have, we recommend backing off to classes of users. As a proof of concept, this paper will form classes of users based on the first few bytes of the IP address. Even better is to back off based on market segments and collaborative filtering (other users who click similarly). Time and geography can be viewed as surrogate variables for demographics in market segmentation analysis.

The model assumes that users in a class share similar interests. For example, users from the same company are likely to ask similar questions and click on similar answers. Consequently, if we lack adequate historical data for a particular user, we can backoff to a larger class of similar users.

Formally, assume a query Q, a user U, and a web document URL. Let $\Gamma = \{C_0, C_1, ..., C_{n-1}\}$ be a set of n classes of users. Under personalization with backoff, the probability $p(URL|Q,U)$ is estimated as a simple linear combination of the class models, for each class that the user is a member of. The weights, λ, can be fit with EM [9]. That is,

$$p(URL|Q,U) = \sum_{C_i \in \Gamma} \lambda_{U,i} p(URL|Q,C_i)$$

where $\sum_i \lambda_{U,i} = 1$, and $\lambda_{U,i} = 0$ if $U \notin C_i$.

Note that the classes need not form a partition. In particular, we will place IP addresses into a nested hierarchy. Each IP address can be a member of multiple nested classes. Certainly, IP hierarchy is not the only possible choice of the user classes, and perhaps not the best choice either.

This model allows for a wide range of personalization. Two extreme special cases are 0% personalization and 100% personalization. We will refer to 0% personalization as non-personalized, and 100% personalization as complete personalization.

Non-personalization (or 0% personalization) is the special case where $n = 1$. There is just one super-class of users: $\Gamma = \{C_0\}$. All users are members of this single super-class. In this special case, the model becomes $p(URL|Q,U) = p(URL|Q,C_0)$, where C_0 can be dropped.

At the other extreme, 100% personalization, $n = |U|$. Every class contains exactly one user, and every user belongs to exactly one class. In this special case, the model becomes $p(URL|Q,U) = p(URL|Q,C_u)$, where $C_u = \{U\}$.

Between these two extreme cases, there is plenty of middle ground, where users are grouped into more than one class, but less than $|U|$. Class assignments are typically determined by variables such as IP addresses, time and geography, and combinations thereof. These variables can be treated as surrogates for demographic variables.

4.2 Nested Classes Based on IP Addresses

Users are assigned to 5 nested classes based on their IP address. It is assumed that the prefix of an IP address is a convenient surrogate for some more meaningful variables such as geography. An IP address consists of four sections, each of which is typically encoded with a byte. This representation suggests the following 5 classes:

- IP_4: Users are assigned to classes based on all 4 bytes of the IP address.

- IP_3: Users are assigned to classes based on the 3 most significant bytes of the IP address.

- IP_2: Users are assigned to classes based on the 2 most significant bytes of the IP address.

- IP_1: Users are assigned to classes based on the most significant byte of the IP address.

- IP_0: All users are assigned to a single super-class.

With this construction, every user is assigned to exactly 5 classes. IP_4 and IP_0 are the two extreme cases mentioned above: 100% personalization and 0% personalization, respectively.

We further simplify the model to use just 5 λ's. That is,

$$\begin{aligned} p(URL|Q,IP) &= \lambda_0 p(URL|Q,C_{IP,0}) + \\ &\quad \lambda_1 p(URL|Q,C_{IP,1}) + \\ &\quad \lambda_2 p(URL|Q,C_{IP,2}) + \\ &\quad \lambda_3 p(URL|Q,C_{IP,3}) + \\ &\quad \lambda_4 p(URL|Q,C_{IP,4}) \end{aligned}$$

where $C_{IP,k}$ is the class of IP addresses that share the most significant k bytes.

For for example, the IP address, 156.111.188.243, belongs to 5 nested classes, namely:

$$C_{IP,4} = \{156.111.188.243\}$$

$$C_{IP,3} = \{156.111.188.*\}$$

$$C_{IP,2} = \{156.111.*.*\}$$

$$C_{IP,1} = \{156.*.*.*\}$$

$$C_{IP,0} = \{*.*.*.*\}$$

There are many ways to fit the λ's. We used the standard Expectation-Maximization(EM) algorithm [9], an iterative procedure which estimates the parameters of the model from the training set, and also finds the λ's that *maximize* the validation set V given the model. On each iteration, we perform both an estimation (E) step as well as a maximization (M) step with the following two updating formulae:

$$z^{(n+1)}_{\langle Q,URL,IP \rangle,i} = \frac{\lambda_i^{(n)} p(URL|Q,C_{IP,i})}{\sum_{k=0}^{4} \lambda_k^{(n)} p(URL|Q,C_{IP,k})}$$

$$\lambda_i^{(n+1)} = \frac{\sum_{\langle Q,URL,IP \rangle \in V} z^{(n+1)}_{\langle Q,URL,IP \rangle,i} C(\langle Q,URL,IP \rangle, V)}{\sum_{\langle Q,URL,IP \rangle \in V} C(\langle Q,URL,IP \rangle, V)}$$

where $p(URL|Q,C_{IP,k})$ denotes probability estimates based on the training set, and $C(\langle Q,URL,IP \rangle, V)$ denotes counts of triples based on the validation set.

Figure 2 shows the resulting estimates of λ. The training set is a month of logs (January 2006). The validation set is a single day of logs (February 1st).

Figure 2 shows that a little bit of personalization is better than too much or too little. Note that λ_3 and λ_2 are considerably larger than λ_4, λ_1 and λ_0. Too much personalization suffers from sparse data whereas too little personalization misses the opportunity of personalization.

Interestingly, we also see that λ_0 is larger than λ_1 and λ_4. We see this because the validation set contains many IP addresses that do not appear in the training set.
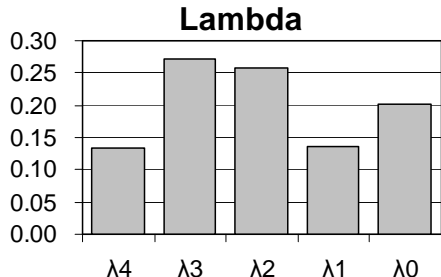
Figure 2: A little bit of personalization is better than too much or too little. Note that λ_2 and λ_3 are larger than the other λ's. Too much personalization (λ_4) runs into sparse data, whereas too little (λ_0 and λ_1) misses the opportunity. The EM algorithm assigns more weight to classes of users that share a few bytes of their IP address than to classes that share more (100% personalization) or less (0% personalization).

4.3 Evaluation of the Backoff Model

How well does this model of personalization perform on future queries? With appropriate smoothing (backoff), personalization should do no harm. Hopefully, personalization improves (reduces) entropy by enough to justify the effort. But no matter what, it should never hurt.

To evaluate the model, we used the logs between February 2, 2006 and February 6, 2006 as a test set, T. We constructed 5 properly nested subsets:

$$T_4 \subseteq T_3 \subseteq T_2 \subseteq T_1 \subseteq T_0 \subseteq T$$

The 5 subsets exclude queries that were not seen in the training set because they could not benefit from this model of personalization. The remainder of the $\langle Q, URL, IP \rangle$ triples in T were assigned to T_0, T_1, T_2, T_3, T_4 based on the most significant k bytes of the IP address. The smallest subset, T_4, contains the triples where all 4 bytes of the IP address were observed in the training set. This set is properly nested within T_3, which contains triples where the first 3 bytes of the of the IP address were observed in training. And so on. Figure 3 shows that T_0, T_1, T_2, T_3, T_4 cover between 8.8% and 51.0% of the triples in T.

Figure 3 shows that our proposal, personalization with backoff (dashed lines), does not harm, as we would hope. That is, the dashed line improves (lowers) cross entropy over the "no personalization" baseline, across all 5 test subsets.

In addition, personalization with backoff beats the "complete personalization" baseline in 4 of the 5 subsets. Obviously, backoff can't beat 100% personalization when you have the relevant data (T_4), but even in that case, backoff isn't much worse.

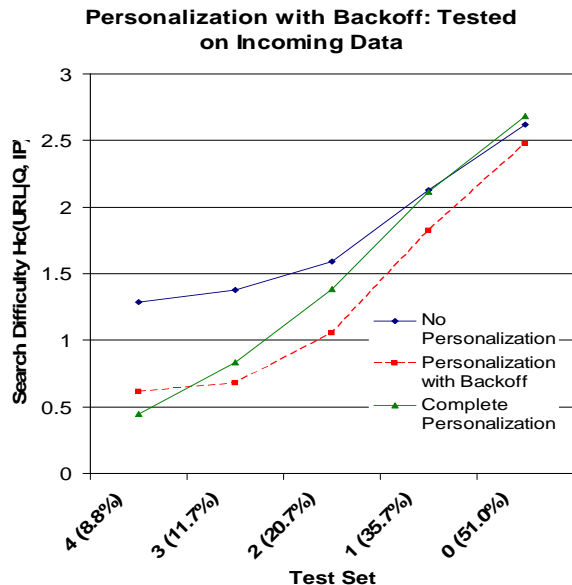This section proposed a novel backoff approach to personalization. Backoff is a classic smoothing technique bor-

Figure 3: In 4 of the 5 test subsets, our proposal, personalization with backoff (dashed lines), has better (lower) cross entropy, $H_c(URL|IP, Q)$, than two baselines: too much personalization (triangles) and too little personalization (solid lines with squares).

rowed from language modeling. We show the effectiveness of personalization with backoff using IP addresses. Users are assigned to nested classes, based on the most significant bytes of their IP address. This approach is just one possible way to approach the lower bound of search difficulty estimated with Shannon's entropy. To build a real personalized search engine, this log-based backoff model has to be combined with other features associated with search engines, such as static rank and content relevance. IP address is by no means the only possible surrogate variable for assigning users to classes. The next section will explore some other possibilities.

5. SEGMENTATION VARIABLES

In addition to IP addresses, there are many other variables that could be used for backing off. The next two subsections will explore day of week and time of day, two variables that have been used to segment telephone traffic into businesses and consumers [8]. Consumers and businesses issue different queries at different times. Different market segments have different needs, and ask different questions at different times.

5.1 Day of Week

In well-understood mature businesses like telephony, it is common to observe large and important dependencies on day-of-week. Volumes are typically higher on weekdays than weekends. Volumes are especially high on Mondays. Friday afternoon is almost a weekend. The Monday after a long weekend is even bigger than a typical Monday. There are strong interactions between these trends and market segments. Businesses tend to do most of their work on business days, whereas consumers tend to be more active during Prime Time television hours.

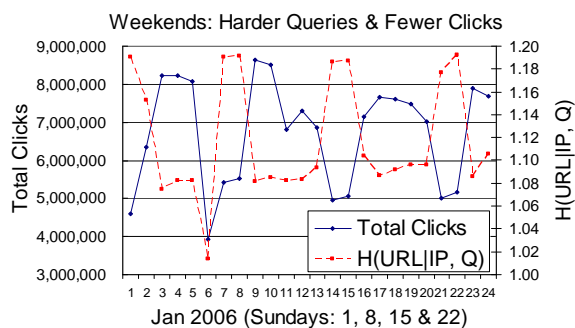Figure 4 shows that there are similar day-of-week patterns

Figure 4: Day-of-week patterns could be used to segment the search market into businesses and consumers. Note that click volumes are out of phase with click entropies.

in the search logs of the first 24 days of January 2006 (where 1/1/06 was a Sunday). As expected, volumes follow the standard pattern, higher during the week than on weekends. Figure 4 also shows entropy by day of week, which is more surprising. The queries on business days are easier than on weekends.

This time structure is very repeatable and robust. Figure 5 reports cross entropy of personalized search. Each of the first 24 days of January 2006 was used as a training set, and each of the first 6 days in February 2006 was used as a test set. All pairs of a testing and a training set are scored by cross entropy: $H_c(URL|IP, Q)$. Some of the pairs used a weekend in training and some didn't. Similarly, some of the pairs used a weekend in testing and some didn't.
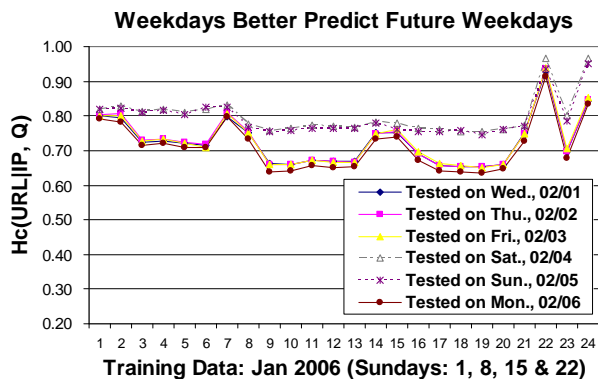

Figure 5: Weekends are harder (more entropy) than business days. Cross entropy peaks both when the weekend is used for training and/or testing.

Weekends are harder (larger entropy) than weekdays. Figure 5 shows six lines, one for each of the 6 test days. Note that the solid lines (business days) are consistently below the dashed lines (weekends). There are 24 points along the x-axis in Figure 5, one for each of the 24 training days. All curves peak on weekends. Weekends are harder, both when used for training as well as testing. From the solid lines, we also learn that future weekdays are better predicted using previous weekdays than using previous weekends.

This analysis suggests that it is potentially beneficial to include the market segmentation of weekdays/weekends along

with IP addresses in personalized search, and treat them accordingly.

5.2 Time of Day

Figure 6 shows query volumes and entropies, $H(Q|IP)$ by hour for the first 15 days of January 2006. There are clear hour of day effects, especially on weekdays.
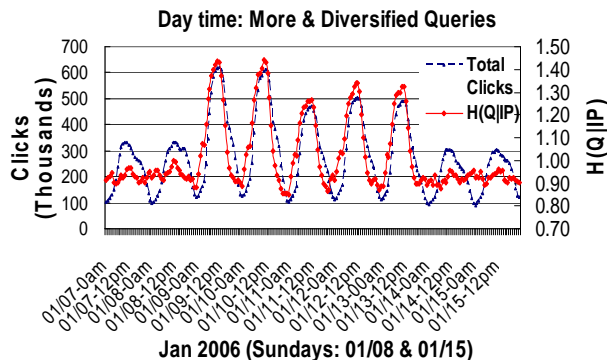

Figure 6: Query volumes and entropies show a clear dependence on hour of day, at least for weekdays.

Volumes follow the expected pattern, with more queries during the day and fewer at night. Yet again, entropy is a surprise. Recall that volumes and entropies were out of phase with one another in Figure 4. This time, they are in phase with one another. It appears that queries are different from clicks.
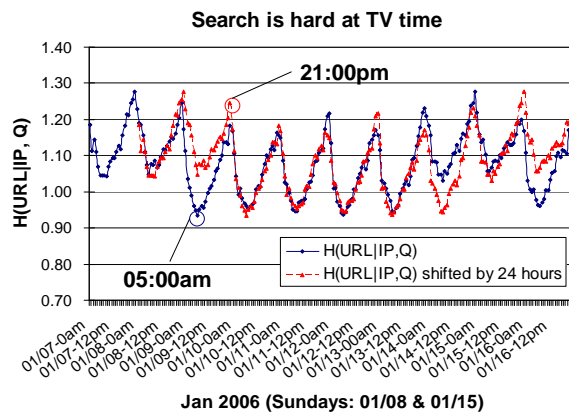

Figure 7: Search is simpler at work hours and more difficult at television hours

Figure 7 shows entropies of clicks, $H(URL|IP, Q)$, by hour from January 7, 2006 to January 16, 2006. There is a strong hourly time structure. Entropy peaks during prime time TV hours. The valleys are very early in the morning.

The dashed line highlights the hourly time structure. The dash line is the solid line shifted right by 24 hours. An autocorrelation analysis would compare these two lines, showing that there is a strong periodicity with a lag of 24 hours, not surprisingly. The plot makes it clear that the daily periodicity is stronger on weekdays than weekends, which again, is not unexpected.

To test whether a market segmentation with time of day could help the personalization of future search activity, we
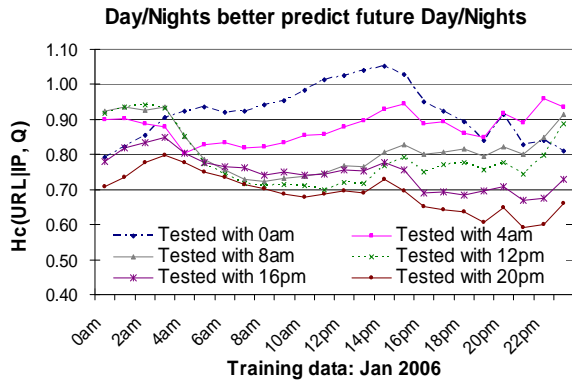
Figure 8: Cross validation of the predictive ability of hours in a day

cross validate the search difficulty with logs during different hours of day as training and testing datasets. Specifically, we partition the search logs in January 2006 into 24 training sets, each corresponding to one hour in the day, and select six hours' search logs on February 1, 2006 as testing sets. The results of $H_c(\text{URL}|\text{IP, Q})$ are plotted in Figure 8. From the plots, we see that the search history during different hours of a day shows different predictive ability over a testing set of different hours. From the two dashed lines, it is easy to see that search in the day time can be better predicted by history of the day time, and nights are better predicted by nights. When the time of the training set is closer to the time of the testing set, the cross entropy becomes lower. When the time of the training set departs from the time of the testing set, $H_c(\text{URL}|\text{IP, Q})$ becomes larger. This suggests that the best training data set to personalize future search activity at a given time of day is the search history at the closest time period of a day.

In this analysis, it is clear that segmenting the search business with the time of day is potentially beneficial on top of IP address and day of week.
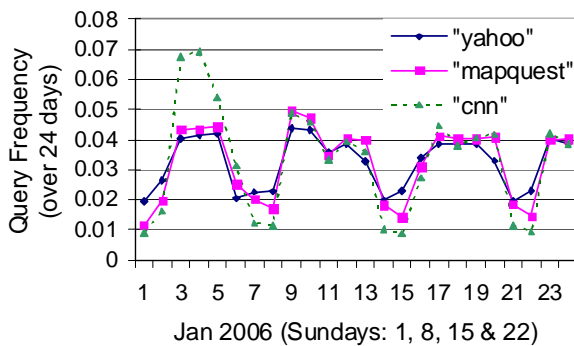
5.3 Query Types

Figure 9: Business queries are issued on business days.

We already presented that market segmentation with geographic variables such as IP addresses, and time variables such as day/week and hour/day are beneficial for personal-

ization. All these variables are metadata variables in search. How about those core variables in search? Is it also beneficial to differentiate the core variable in search, the query?

Figure 9 and Figure 10 present the day-of-week frequency patterns for two groups of queries. The first group includes three query strings "yahoo," "mapquest," and "cnn," and the second group includes "sex," "mp3," and "movies." From Figure 9, we see that the frequency of the first group of queries, which we shall call business queries, has clear day-of-week patterns. There are significantly more business queries on weekdays than weekends. The second group of queries, which we shall call consumer queries, however, does not show clear day-week patterns. As in Figure 10, the frequency of consumer queries on weekends is comparable, sometimes even higher than their frequency on weekdays. It is interesting (but not unexpected), that the query "movies" is asked most frequently on Fridays.
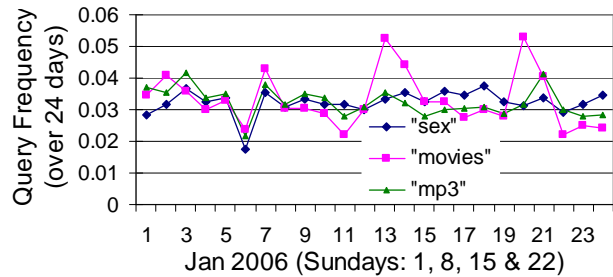

Figure 10: Unlike business queries, consumer queries do not select for business days.

Figure 11 presents the comparison of the time-of-day patterns of the frequency of two queries, "yahoo" and "sex". It is clear that both queries have clear time-of-day patterns. However, the frequency of these two queries gets highest and lowest at different hours.

This analysis shows that besides metadata variables such as geography and time, the search business can also potentially benefit from segmenting the search space with the type of core search variables, such as the queries.
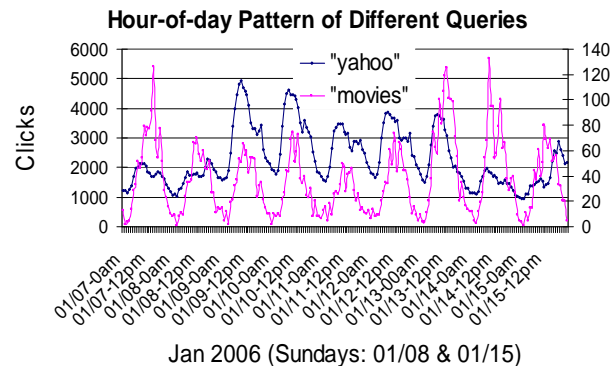

Figure 11: Different types of queries have different hour-day patterns

In general, users ask more questions and simpler questions during business days and business hours. It isn't clear why

this is so, but we might hypothesize that businesses are more business-like, more likely to ask direct questions that have direct answers, like navigational queries. "cnn" is an example of a navigational query. The answer to the "cnn" query is simply: "www.cnn.com." In contrast, consumers ask less direct questions. They may be seeking employment, health, wealth, happiness, entertainment, etc. They may be shopping or just browsing, with no particular place to go, and lots of time on their hands. In extreme cases, one can even find Eliza-like[10] queries in the logs.

6. RELATED WORK

This paper draws connections across a wide range of fields including: Information Theory, backoff smoothing of language models, query suggestions, personalization and marketing. There is a huge body of work in each of these areas, though relatively little work that connects all of them (or even many of the pairs).

Entropy has a long history dating back to Shannon, and perhaps, earlier. See [11] for an excellent retrospective on Shannon's life, work and impact. Entropy has been applied to many data sources, though there is still plenty of room for novel applications such as web logs.

There is a considerable body of work on estimating the size of the web including: [18, 2, 19, 20, 10, 12]. These references attempt to estimate supply (the size of the reachability graph of links from one url to the next), as opposed to demand (clicks). By taking demand into account, we can come up with much more feasible estimates, millions not billions, suggesting that a small cache of a few million pages could capture much of the value.

Entropy analysis is once again well accepted in Computational Linguistics. Back in the late 1940s and early 1950s, Shannon's Information Theory was having a dramatic impact on a wide range of fields from Engineering to Psychology and more. Shannon published his remarkable estimate of the size of English language in 1951 [24]. Chomsky's *Syntactic Structures* [4] came out shortly thereafter in 1957, arguing that language was unbounded (infinite) and that ngram approximations (such as Shannon's) do not come closer and closer to the truth. Chomsky's work dominated much of the thinking in Computational Linguistics over the next few decades, but Information Theory regained popularity in Computational Linguistics in the 1990s with the successes of trigram language models in speech recognition [6]. The speech application motivated researchers to think about smoothing methods such as backoff: [17, 5, 3, 31].

Query suggestions [7, 15, 16, 30] and personalization [14, 13, 27, 29, 25, 26] are somewhat related topics, though the connection between those two topics and backoff smoothing of language models is novel. Many personalized search techniques have been proposed, both server-side [29][11] and client-side [26, 28], as well as with long term query history [29, 28] and short term implicit feedback [25, 21]. Much of this work takes advantage of search engine query logs. This work tends to be focused more on methods of improving user experience, and less on sizing challenges and opportunities.

Market segmentation (and demographics) come from a completely different tradition than Language Modeling and backoff. Marketing is relatively more central to this con-

ference than Information Theory and Language Modeling. Pregibon and Cortes [8], for example, were concerned with marketing applications in telephony. Marketing was eager to find ways to segment customers based on usage (demand). Pregibon and Cortes found that businesses and consumers make calls at different times for different purposes. Marketing could take advantage of such insights to target offers more appropriately since businesses and consumers respond differently to different offers such as various pricing plans and discounts. The connection between, marketing, a well-established KDD application, and Language Modeling is novel.

7. CONCLUSIONS

In this paper, we showed how entropy can be used to address a number of fundamental questions in web search. Entropy was estimated from search logs, a sequence of triples: $\langle Q, URL, IP \rangle$, indicating that a click was observed from a particular URL and a particular IP address in response to a particular query Q.

How big is the web? Answer: millions, not billions.

When the web eventually saturates the market, then the number of home pages, businesses and products will be bounded by $O(population)$. However, unlike telephony, the web is a growth business, far from saturation. For the foreseeable future, we will be able to find millions of famous people and academics, but not everyone (not billions of ordinary people like our neighbors).

Large investments in clusters in the cloud could be wiped out if someone found a way to capture much of the value of billions with a small cache of millions.

While there are lots of pages out there (infinitely many, in a Chomskian sense), there are not that many pages that people actually go to. Shannon's entropy (H) is a powerful tool for sizing challenges and opportunities.

How hard is search? It takes around 22 bits to guess the next url (or the next query or the next user). 22 bits is millions, not billions. When we give the search engine the query, we cut the 22 bits down to around 3 bits.

What is the opportunity for personalization? Personalization cuts the search space in half (from 3 down to less than 1.5 bits). That is a huge opportunity. A personalized cache is an even bigger threat to the cluster in the cloud than a cache without personalization.

Shannon's entropy provides a novel way to think about sizing challenge and opportunity in search business. It gives the lower bound of the difficulty of personalized search but not an operational procedure to approach this lower bound. In reality, when we do not have data for a particular user, a smoothed version of the language model $P(URL|Q, U)$ has to be applied. While different smoothing methods lead to different personalization approaches, we introduce one of those choices: personalization with backoff.

Personalization with backoff is more effective than personalization without backoff. As a proof of concept, we discussed backing off to classes of users based on IP addresses. A little bit of personalization was found to be better than too much or too little.

Rather than backing off based on prefixes of IP addresses, it would be better to back off based on market segmentation (demographics) and/or collaborative filtering (other users who click like you). Different segments have different needs (ask different questions at different times) and

---

[10]http://en.wikipedia.org/wiki/ELIZA
[11]See also www.google.com/psearch.

different values (willingness to pay and advertising opportunities). Businesses and consumers ask different questions at different times. We showed that query volumes and search difficulty (entropy) vary by time of day and day of week. Variables such as IP addresses and time and geography can be viewed as convenient surrogates for more sensitive market segmentation variables.

There are many possible future extensions to this work. It is interesting to introduce alternative principled smoothing methods, probably with backing off based on combinations of demographic variables. We are particularly excited by the possibility of backing off based on collaborative filtering (other users with similar search interests). It is interesting to combine the language model of personalization with backoff with other well known features in search, build a real personalized search engine, and evaluate the effectiveness of personalization with real user experiments.

8. ACKNOWLEDGEMENTS

9. REFERENCES

[1] C. Anderson. *The Long Tail: Why the Future of Business is Selling Less of More.* Hyperion, 2006.

[2] K. Bharat and A. Broder. A technique for measuring the relative size and overlap of public web search engines. In *Proceedings of the seventh international conference on World Wide Web 7*, pages 379–388, 1998.

[3] S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 310–318, 1996.

[4] N. Chomsky. *Syntactic Structures.* The Hague/Paris: Mouton, 1957.

[5] K. Church and W. Gale. A comparison of the enhanced good-turing and deleted estimation methods for estimating probabilities of english bigrams. *Computer Speech and Language*, 5(1):19–54, 1991.

[6] K. Church and R. Mercer. Introduction to the special issue on computational linguistics using large corpora. *Computational Linguistics*, 19(1):1–24, 1993.

[7] K. Church and B. Thiesson. The wild thing! In *Proceedings of the ACL 2005*, pages 93–96, 2005.

[8] C. Cortes and D. Pregibon. Signature-based methods for data streams. *Data Min. Knowl. Discov.*, 5(3):167–182, 2001.

[9] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statist. Soc. B*, 39:1–38, 1977.

[10] D. Dhyani, W. K. Ng, and S. S. Bhowmick. A survey of web metrics. *ACM Comput. Surv.*, 34(4):469–503, 2002.

[11] R. Gallager. Claude E. Shannon: A retrospective on his life, work, and impact. *IEEE Transactions on Information Theory*, 47(7):2681–2695, 2001.

[12] A. Gulli and A. Signorini. The indexable web is more than 11.5 billion pages. In *Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 902–903, 2005.

[13] G. Jeh and J. Widom. Scaling personalized web search. In *Proceedings of the 12th international conference on World Wide Web*, pages 271–279, 2003.

[14] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142, 2002.

[15] R. Jones and D. C. Fain. Query word deletion prediction. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 435–436, 2003.

[16] R. Jones, B. Rey, O. Madani, and W. Greiner. Generating query substitutions. In *Proceedings of the 15th international conference on World Wide Web*, pages 387–396, 2006.

[17] S. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speeech and Signal Processing*, 35(3):400–401, 1987.

[18] S. Lawrence and C. L. Giles. Searching the World Wide Web. *Science*, 280(5360):98–100, 1998.

[19] S. Lawrence and C. L. Giles. Accessibility of information on the web. *Nature*, 400(6740):107–109, 1999.

[20] S. Lawrence and C. L. Giles. Accessibility of information on the web. *Intelligence*, 11(1):32–39, 2000.

[21] F. Radlinski and T. Joachims. Query chains: learning to rank from implicit feedback. In *Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 239–248, 2005.

[22] C. Sagan. *Billions and Billions: Thoughts on Life and Death at the Brink of the Millennium.* New York: Random House, 1997.

[23] C. Shannon. A mathematical theory of communication. *Bell Systems Technical Journal*, 27:379–423, 623–656, 1948.

[24] C. Shannon. Prediction and entropy of printed english. *Bell Systems Technical Journal*, 30:50–64, 1951.

[25] X. Shen, B. Tan, and C. Zhai. Context-sensitive information retrieval using implicit feedback. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 43–50, 2005.

[26] X. Shen, B. Tan, and C. Zhai. Ucair: a personalized search toolbar. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 681–681, 2005.

[27] K. Sugiyama, K. Hatano, and M. Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In *Proceedings of the 13th international conference on World Wide Web*, pages 675–684, 2004.

[28] B. Tan, X. Shen, and C. Zhai. Mining long-term search history to improve search accuracy. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 718–723, 2006.

[29] J. Teevan, S. T. Dumais, and E. Horvitz. Personalizing search via automated analysis of interests and activities. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 449–456, New York, NY, USA, 2005. ACM Press.

[30] R. W. White and G. Marchionini. Examining the effectiveness of real-time query expansion. *Information Processing and Management (IPM)*, 43(3):685–704, 2007.

[31] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 334–342, 2001.