Intention Modeling for Web Navigation

Xiaoming Sun
Dept. of Computer Science & tech., Tsinghua Univ., Beijing 100084, China
sun_xm97@mails.tsinghua.edu.cn

Zheng Chen
Microsoft Research Asia, 49 Zhichun Road, Beijing 100080, China
zhengc@microsoft.com

Liu Wenyin
Dept. of Computer Science, City University of Hong Kong, Hong Kong, China
csliuwy@cityu.edu.hk

Wei-Ying Ma
Microsoft Research Asia, 49 Zhichun Road, Beijing 100080, China
wyma@microsoft.com

Abstract

A novel global optimization method referred to as the multi-step dynamic n-gram model is proposed for predicting the user's next intended action while he/she is surfing the Web. Unlike the traditional n-gram model, in which the predicted action is taken as the ultimate goal and is only determined by its previous n actions, our method predicts the next action that lies on the optimal path that leads to the ultimate goal. Experiments show that the prediction accuracy of our proposed method can achieve up to 3.65% (or about 11% relative) improvement to the traditional one-step n-gram model.

1. Introduction

The Internet has been growing in an incredible speed. It was reported that in the year 1999 there were at least 9 million web servers and 1.5 billion web pages on the Internet. Some experts had estimated that this number would reach 7.7 billions by the end of the year 2001. Facing such a huge "database" the user might easily lose his way even with the help of search engines. Another problem that the user is facing is the accessing speed of the Internet. Although broadband networks have been deployed in many places, the time delay of information transportation on the Internet is still a serious problem. Various pre-fetching techniques have been introduced to deal with the problem, in which the user's intended information is predicted and pre-fetched to nearby caches before the user actually requests them. Because this process should be performed in real-time, there is a need to detect the user's real intentions as quickly and precisely as possible while he/she is surfing on the Web. In other words, the system has to predict which hyperlink is the one that the user really wants to follow. In this paper, we focus on how to predict the user's intention from his/her "Web surfing history" (the page sequences that the user had ever visited).

The prediction of the user's intention can be used in many ways to help the user surf the Web with better experience. The first one is that we can easily recommend several related or similar hyperlinks to the user as done by Balabanovic [Bal97]. The second application of user intention modeling is for pre-fetching. Those web pages that are potentially interesting to the user can be pre-fetched into the cache for future usage. Padmanabhan etc. [PaM96] applied the pre-fetching method to cache the web pages that are likely to be requested soon. By this way it could reduce the latency perceived by the user, saving the user's time. The third application of user intention modeling is for website structure optimization. The editor can reorganize the hyperlink's structure of a web site based on the analysis of their users' intentions.

Currently, there are many systems and agents, including WebWatcher [JFM97], WebMate [ChS98], and so on, which are trying to predict the user's navigation intention from the user's previous navigation paths. The methods and models that have been used by these systems include content-based methods, Markov chain models, Bayesian Networks model and path-based methods. We will give a detail review of these methods in Section 2. Unfortunately, most of these systems only consider one step forward. Since the web pages that each user has visited are very limited, the visited pages are very sparse in the data space. Hence, in many cases the prediction results are only local optimal. In this paper, we propose a new method which considers multiple steps forward while dynamically applying the n-gram model to discover the user's real intention. It is a global optimization method for user intention prediction compared to the one-step n-gram model. Obviously, this method is difficult to implement. Hence, we simplify the proposed method such that it can be easily implemented. One of the research problems is to determine how many steps we can predict. We apply the entropy evaluation method to select the most appropriate prediction steps in our implementation.

The rest of this paper is organized as follows. In Section 2 we review some related works on user's intention prediction. In Section 3 we present in detail our multi-step dynamic n-gram model and its implementation. In Section 4 we present our experiment's results of applying the proposed method. Finally, we present concluding remarks in Section 5.

2. Related work

Webwatcher [JFM97] is a very famous recommendation system that helps users navigate the Web such that they can quickly find their desired or interested information. The system used traditional information retrieval methods (such as TF*IDF) to evaluate the similarity between two documents, and applied a reinforcement learning method to the website structure to assist the user navigate the Web. Albrecht et al. [AZN99] built a hybrid Markov model which combined four Markov models for pre-fetching documents. They assumed that the page sequence a user had visited was a Markov chain and applied the time factor in the Markov model. Lau and Horvitz [LaH99] built a Bayesian Network to predict user's next query using only query and trim information. They assumed that the next query only depended on the previous query and the time interval, and is independent on other factors. Pitkow et al [PiP99] built a path-based system. They wanted to find out the longest repeating page subsequence (a path) that all users have visited. Su et al. [SYL00] applied the *n*-gram language model into the pre-fetching systems. They considered a sequence of *n* web pages as an *n*-gram. By counting the times each *n*-gram appears, they give the prediction based on the maximal count.

Our work is also based on the path-based model and does not consider the page's content. Compared to the works of Pitkow et al [PiP99] and Su et al. [SYL00], we used the probability instead of the *n*-gram count. Most importantly, we propose a multi-step dynamic n-gram model to predict several steps ahead such that the ultimate web-page is the user's real intention. Furthermore, we also apply several other models, which have been successfully used in speech recognition and other domains, to improve the prediction accuracy.

3. Statistical Prediction Models

The user's intention prediction is based on the user's historical navigation paths. *N*-Gram probability model is a very efficient statistical method in natural language processing and automatic speech recognition. In this paper, we apply the tri-gram probability model (one of the popular n-gram model) as our baseline and compare it with our proposed global optimization method. Next, we first briefly describe the n-gram model.

3.1. N-gram model

Usually, the user's navigation path can be represented as a sequence of visited web pages $w_1, w_2, \ldots, w_i, \ldots, w_L$, where w_i is the *i*th visited web-page in the sequence. In order to estimate the probability of the navigation path, we apply the Bayesian rule to rewrite the probability estimation as Eq. (1).

$$\Pr(w_1, w_2, \ldots, w_i, \ldots, w_L) = \Pr(w_1)\prod_{i=2}^{L}\Pr(w_i \mid w_1, \ldots, w_{i-1}) \quad (1)$$

We applied the statistical language model (SLM) to estimate the probability $\Pr(w_i \mid w_1, \ldots, w_{i-1})$ in Eq. (1). The most widely used statistical language model is the so-called n-gram Markov model [FrJ97]. The n-gram language model assumes that each word in the sequence is only determined by its previous (*n*-1) words. That is, Pr(wi|w1…wi-1) = Pr(wi|wi-n+1…wi-1). Similarly, in this paper each web page sequence with length *n* was called an *n*-gram web-page sequence. We assume that the next hyperlink the user will click is only dependant on the previous (*n*-1) hyperlinks user has just clicked. Hence, the n-gram probability is re-written in Eq. (2).

$$\begin{aligned}\Pr(w_i \mid w_1, \ldots, w_{i-1}) &\approx \Pr(w_i \mid w_{i-n+1}, \ldots, w_{i-2}, w_{i-1}) \\ &= \frac{\Pr(w_{i-n+1}, \ldots, w_{i-2}, w_{i-1}, w_i)}{\Pr(w_{i-n+1}, \ldots, w_{i-2}, w_{i-1})} \\ &= \frac{C(w_{i-n+1}, \ldots, w_{i-2}, w_{i-1}, w_i)/C_n}{C(w_{i-n+1}, \ldots, w_{i-2}, w_{i-1})/C_{n-1}} \\ &= \frac{C(w_{i-n+1}, \ldots, w_{i-2}, w_{i-1}, w_i)}{C(w_{i-n+1}, \ldots, w_{i-2}, w_{i-1})} * C\end{aligned} \quad (2)$$

where, $C(w_{i-n+1}, \ldots, w_{i-2}, w_{i-1}, w_i)$ denotes the count of the *n*-Gram $(w_{i-n+1}, \ldots, w_{i-2}, w_{i-1}, w_i)$ appearing in the training data. C_n is the total number of the *n*-grams. C_{n-1} is the total number of the (*n*-1)-grams. C equals to C_n/ C_{n-1}. C_n, C_{n-1}, and C are constants. From equation **(1)** and **(2)** we know that if $C(w_{i-n+1}, \ldots, w_{i-2}, w_{i-1})$ is known, the probability of $\Pr(w_i \mid w_1, w_2, \ldots, w_{i-1})$ is only influenced by the count $C(w_{i-n+1}, \ldots, w_{i-1}, w_i)$.

In this paper, we use the tri-gram model (n=3) as our baseline. It has been proved that within a large training corpus tri-gram works better than bi-gram (n=2). Once a tri-gram (w_{i-2}, w_{i-1}, w_i) does not exist in the training data, we back-off it to the bi-gram model (w_{i-1}, w_i), i.e.,

$$\Pr(w_i \mid w_{i-2}, w_{i-1}) = \alpha(w_{i-2}) \Pr(w_i \mid w_{i-1}),$$

where $\alpha(w_{i-2})$ is the back-off weight [FrJ97]. Furthermore, the smoothing technology [FrJ97] is also applied to deal with the data sparseness problems.

3.2. Global Optimization Model

As we mentioned in Section 1, our goal is to find out the user's real intention based on the user's previous behaviors. However, unlike most current methods, which just predict one step ahead, we predict several steps ahead such that the ultimate web-page after several steps is the user's real intention. Suppose the user has already visited k-1 web-pages $w_1, w_2, \ldots, w_i, \ldots, w_{k-1}$, and the user's real intention is w_L. Our goal is to find out the path $w_k, w_{k+1}, \ldots, w_L$ such that the probability of the overall navigation path $\Pr(w_1, w_2, \ldots, w_{k-1}, w_k, \ldots w_L)$ is maximized. This is a global optimization method. The one-step n-gram model made an assumption that the local optimization at the next step is the user's real intention. Thus it only chooses a w_k to maximize the probability $\Pr(w_1, w_2, ..., w_{k-1}, w_k)$ instead of the global probability. Although this one-step optimization is an efficient solution, it is likely to reach a local optimal point, especially when the data is not sufficient. This is similar to the "hill-climbing" algorithm and other searching algorithms in AI. For example, if a user likes to read news on a news web-site but his/her favorite part of news is in always at a very deep level (e.g., the fourth level), each time the user must follow three hyperlinks to reach it. The user is not interested in all of the hyperlinks that are included in this path excepting the last one. In this case, the hyperlinks at the beginning of this path may have very small probabilities. Thus if we use one-step n-gram prediction, the first predicted hyperlink might take the user to a wrong way and he/she would never arrive his/her goal.

In order to avoid reaching the local optimal, we try to maximize the probability of the entire path instead of predicting only one step as mentioned in Section 3.1. Our global optimization method is formulated in Eq. (3).

$$\mathop{\arg\max}_{w_k} \prod_{i=k}^{\infty} \Pr(w_i \mid w_1 ... w_{i-2} w_{i-1}) \quad (3)$$

Next, we prove that it reflects the probability of the entire path, i.e. $\Pr(w_{k+1} w_{k+2} ... \mid w_1 ... w_{k-1} w_k)$. The proof is shown in Eq. (4).

$$\begin{aligned}
&\prod_{i=k+1}^{L} \Pr(w_i \mid w_1 ... w_{i-2} w_{i-1}) \\
&= \prod_{i=k+1}^{L} \frac{\Pr(w_1 ... w_{i-2} w_{i-1} w_i)}{\Pr(w_1 ... w_{i-2} w_{i-1})} \\
&= \frac{\prod_{i=k+1}^{L} \Pr(w_1 ... w_{i-2} w_{i-1} w_i)}{\prod_{i=k+1}^{L} \Pr(w_1 ... w_{i-2} w_{i-1})} \\
&= \frac{\Pr(w_1 ... w_{L-1} w_L)}{\Pr(w_1 ... w_{k-1} w_k)} \\
&= \Pr(w_{k+1} ... w_{L-1} w_L \mid w_1 ... w_{k-1} w_k)
\end{aligned}$$

(4)

Hence, if we let L goes to infinite, we can obtained the desired global optimization result.

Furthermore, if we assume the process of user navigation is a second-order Markov process [Fel71],

$$\mathbf{Pr}(w_i \mid w_1 ... w_{i-2} w_{i-1}) = \mathbf{Pr}(w_i \mid w_{i-2} w_{i-1}) .$$

So we can simplify Eq. (3) to Eq. (5):

$$\arg\max_{w_i} \prod_{i=k+1}^{\infty} \mathbf{Pr}(w_i \mid w_{i-2} w_{i-1}) \quad (5)$$

Although Eq. (5) is a simplified model, its complexity is still very high and further approximations should be made for a practical implementation. We propose to use a dynamic multi-step prediction method to reduce the complexity. In real implementation, we only calculate from i=*k+1* to *i=k+t* in Eq. (5), where *t* is a parameter representing that how many steps should be predicted forward. *t* is determined dynamically as follows. We employ the perplexity [FrJ97] to measure the efficacy of the *t*-steps prediction. It reflects the entropy of the path. The perplexity of t-steps prediction is defined in Eq. (6)

$$\frac{1}{t} \sum_{i=k+1}^{i=k+t} \log(\mathbf{Pr}(w_i \mid w_{i-2} w_{i-1})) \qquad (6)$$

Finally the optimal goal can be written in Eq. (7).

$$\arg\max_{t} \left(\frac{1}{t} \sum_{i=k+1}^{i=k+t} \log \mathbf{Pr}(w_i \mid w_{i-2} w_{i-1})\right) \quad (7)$$

4. Experiments

We have used the same **NASA** data set in our experiments as Su et al. had [SYL00]. The **NASA** data set contains about two months HTTP requests to the **NASA Kennedy Space Center** Web Server. In general, a web log can be regarded as a sequence of user requests. Each entry includes the user id, user request and the starting time. It is easy to divide the log file by the user id. Furthermore we can divide each user's logs by the time interval between two requests. We call such a request set a session. We used about 80% sessions as training data and the rest 20% as testing data.

At first we compare the average prediction accuracies of the two models and show them in Figure 1. The first bar is the prediction accuracy of one-step tri-gram prediction model. The second bar is the global optimization model described in Eq. (7). As can be seen from Figure 1, the multi-step global optimization method outperforms the one-step n-gram model by 1% (or with a 2.7% relative improvement).

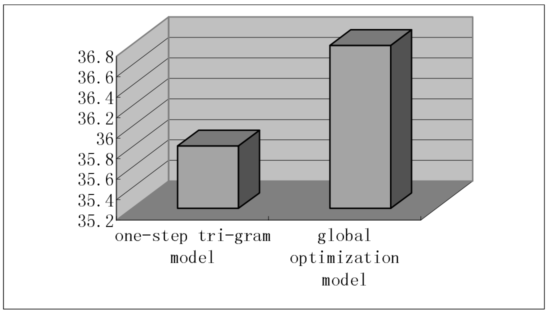

Figure1 Prediction accuracy comparison among different models

In order to prove that the global optimization constantly outperforms the n-gram model, we do the t-test experiments of the two models. We randomly split the entire data set into five pieces. In each of the five experiments we select four pieces as the training data and the rest piece as the testing data. The prediction accuracy of the five experiments is listed in Table 1.

Table1 Performance comparison between the tri-gram model and the global optimization model in the t-test

Experiment ID	1	2	3	4	5
One-step tri-gram prediction	37.29 %	37.23 %	37.25 %	37.35 %	37.24 %
global optimization prediction	37.96 %	38.14 %	37.79 %	38.02 %	38.01 %

The paired-samples t-test with two-tailed distribution is applied to test the effect of our proposed method. The P value of the t-test [Fel71] is 0.00032, which shows that the global optimization prediction has a significant improvement (when P value of the t-test is smaller than 0.05 [Fel71]) compared to the traditional one-step tri-gram prediction model.

All of the above experiments do not include the prediction of the action “stop” (which is the action that the user wants to terminate the session). In the experiments of predicting the “stop” action, the accuracy of the one-step tri-gram model is about 31.69%, and our multi-step global optimization model’s accuracy is 35.34%, which is a 3.65% (or about 11% relative) improvement to the one-step tri-gram model.

5. Conclusion

In this paper we presented a new method for predicting the user’s browsing intention based on the web page sequences he had previously visited. The proposed method is a global optimization method, which employ a multi-step dynamic n-gram model. Compared with the well-known one-step tri-gram model, our proposed model achieved better performance in all situations. The experiments have shown a substantial (up to 3.65% or about 11% relative) improvement, while the P value of t-test is 0.00032, which is smaller than 0.05 and means the improvement is significant [Fel71].

Reference

[AZN99] Albrecht, D.W., Zukerman, I., and Nicholson, A.E. (1999) Pre-sending documents on the WWW: A comparative study. In: *Proc. of the Sixteenth International Joint Conference on Artificial Intelligence* (IJCAI99)

[Bal97] Balabanovic, M. (1997) An adaptive web page recommendation service. In: *Proc. of the First International Conference on Autonomous Agents*, 1997.

[BDD92] Brown, P. F., Della Pietra, V. J., deSouza, P. V., Lai, J. C., & Mercer, R. L. (1992) Class-Based n-gram Models of Natural Language. *Computational Linguistics*, 18(4), 467-479.

[BuS84] Buchanan,B.G., and E.H. Shortliffe. (1984) *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*, Reading, Mass.: Addison-Wesley.

[ChS98] Chen L. and Sycara K. (1998) WebMate: A Personal Agent for Browsing and Searching. In: *Proc. of the Second International Conference on Autonomous Agents*, pp. 132-139.

[CoT91] Cover, T. and Thomas, J. (1991). *Elements of Information Theory*. John Wiley & Sons, New York, NY

[Fel71] Feller, W. (1971) *An introduction to Probability Theory and its applications, volume II*. Wiley, second edition.

[FrJ97] Frederick J. 1997. *Statistical Methods for Speech Recognition*. The MIT Press, Cambridge, Massachusetts.

[JFM97] Joachims, T., Freitag, D., and Mitchell, T. (1997) WebWatch: A tour guild for the World Wide Web. In: Proc. of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI 97), – pp. 770-775.

[LaH99] Lau T., and Horvitz, E., (1999) Patterns of search: analyzing and modeling web query refinement. *In: Proc. User Modeling ’99*, pp. 119-128.

[PaM96] Padmanabhan, V.N. and Mogul, J.C. (1996) Using predictive prefetching to improve World Wide Web latency. *ACM Computer Communication Review*, 26(3), pp. 22-36

[PiP99] Pitkow J. and Pirolli P. (1999) Mining Longest Repeating Subsequences to Predict WWW Surfing. In: *Proc. of the 1999 USENIX Annual Technical Conference*.

[SYL00] Su, Z., Yang Q., Lu Y., Zhang, H (2000) WhatNext: A Prediction System for Web Requests using N-gram Sequence Models. In: *Proc. of the International Conference on Web Information Systems Egineering (WISE2000).*