Focused Crawling by Learning HMM from User's Topic-specific Browsing

Hongyu Liu[1], Evangelos Milios[1], Jeannette Janssen [1,2]
[1]Faculty of Computer Science, Dalhousie University
{hongyu,eem}@cs.dal.ca
[2]Dept. of Mathematics and Statistics, Dalhousie University
janssen@mathstat.dal.ca
Halifax, NS, Canada B3H 1W5

Abstract

A focused crawler is designed to traverse the Web to gather documents on a specific topic. It is not an easy task to predict which links lead to good pages. In this paper, we present a new approach for prediction of the important links to relevant pages based on a learned user model. In particular, we first collect pages that a user visits during a learning session, where the user browses the Web and specifically marks which pages she is interested in. We then examine the semantic content of these pages to construct a concept graph, which is used to learn the dominant content and link structure leading to target pages using a Hidden Markov Model (HMM). Experiments show that with learned HMM from a user's browsing, the crawling performs better than Best-First strategy.

1. Introduction

With the exponential growth of information on the World Wide Web, there is great demand for developing efficient and effective methods to organize and retrieve the information available. According to the study [4], in 2003 the surface web is about 7.5 billion pages and the deep web is more than 5500 billion. However, Google currently indexes about 4 billion web pages, in other words, Google – which has the greatest coverage of all search engines – only covers very small portion of the publicly accessible web, and the other major search engines do even worse.

Due to the imbalance between the exploding volume of the Web and limited storage resources, search engines should attempt to download the high quality pages and include (only) them in their index — a page cannot be retrieved if it has not been indexed. Focused crawling is increasingly seen as a potential solution. It is designed to traverse a subset of the Web to only gather documents on a specific topic and aims to identify the promising links that lead to target documents, and avoid off-topic branches.

Previous work in focused crawling includes [2, 1, 5, 7]. A framework to evaluate different crawling strategies is described in [6]. They found that the Best-First strategy performed best. An interesting approach proposed in [3] is using backlinks to construct context graphs. However, it is infeasible for a focused crawler to rely on search engines like Google to obtain backlink information. Furthermore, due to rapidly growing and mixing various topics in the Web graph, the assumption that all pages in a certain level from a target document will share terms does not always hold.

By contrast, our approach is to use a combination of semantic content analysis and dominant link structure leading to targets to learn the user's browsing patterns by HMM and emulate them to find more relevant pages. After training, our system may have patterns like "University pages are more likely to lead to Research papers than Sports pages". Note this is different from other systems in that we capture semantic associative relations rather than physical link distances to make predictions. The learned model is a general model and parameters of the model are adapted to different users searching different topics, so it is broadly applicable. It can be used to build topic portals and it can help individual users perform personalized online search.

2 System Architecture

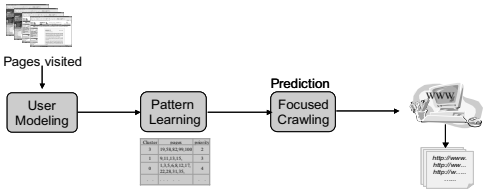

Figure 1. System Architecture.

Fig.1 shows the system architecture. The system consists of three stages: User Modelling, Pattern Learning and Focused Crawling.

2.1 User Modelling

In this stage, we aim to collect the sequences of pages visited by the user during her topic-specific browsing. If user finds current browsing page is interesting, she can click the *Useful* button which is added to page being browsed,

then the page will become an annotated target page. In order to analyze the user browsing pattern, including not only content of pages, but also link structure, we construct a *web graph* which contains both nodes and edges. As Fig.2 shows, each node presents a HTML page with a unique URL, and an edge is established if a referring document has a hyperlink pointing to a referred document. Marked target pages are indicated as double(red)-circled nodes in the web graph.

Figure 2. User Modelling.

The goal of considering only visited pages that have been marked as useful in defining a web graph is to reject some noisy hyperlinks that may mislead the focused crawler, and to extract the dominant structure the user follows towards her topic. The model lets users annotate interesting web pages, and user interests are not restricted to one topic during browsing session.

2.2 Learning the User's Topics

Given the labelled training data from the User Modelling stage, the next stage is to learn user's browsing patterns towards her topics.

Document Representation – Identify Semantic Content

When a user browses on a topic, the topic can be characterized by words contained in web pages on the topic. Different words with similar meanings may be used in pages on the same topics, so we bring context into consideration by representing documents using Latent Semantic Indexing(LSI) [8] to exploit the semantic relationships among different words on the basis of co-occurrences in the document collection. Its effectiveness has been demonstrated empirically in many information retrieval applications leading to increased average retrieval precision. The main drawback of this technique is that the computation cost for large document collections is high. However, in our system, user visited pages form a small collection with limited topic categories, thus LSI appears to be an a good choice.

Document representation is computed as follows:

1. Extract words from user visited page collection, remove stop words and form dictionary of the collection;

2. Calculate its normalized TF-IDF (Term Frequency, Inverse Document Frequency) representation;

3. Apply LSI to obtain a low dimensional LSI space representation.

Clustering – Capture Associative Relations

In our system, we first use K-Means algorithm to cluster

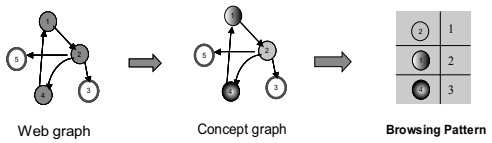

Figure 3. Pattern Learning.

training documents into 5 clusters (K=5) with cosine document similarity metric. The choice of K is based on the assumption that the pages the user visits belong to a limited number of distinct topics when she is focused on topic-specific browsing rather than arbitrary surfing. It is these topics that clustering tries to capture. To find the model that best fits the data, we further apply EM algorithm to optimize the quality of these 5 clusters. The goal is to be able to capture the concept associative relation between different types of documents, i.e., document type A is more likely to lead to targets than document type B, and we don't need to know what exact categories they belong to.

After clustering, the associative relationships between groups are captured into a *concept graph*, where each node is assigned the label of the cluster, as shown in Fig.3.

2.3 Learning Patterns Leading to Targets

We propose a statistical method which uses HMM to estimate the likelihood of topics leading to a target topic directly or indirectly. Hidden Markov Models(HMMs), widely used in speech-recognition, provide superior pattern recognition capabilities for dynamic patterns. HMMs are useful when one can think of underlying unobservable events probabilistically generating surface events, that are observable. In our system, we assume that there is an underlying Markov chain of hidden states, defined as the number of hops away from targets, from which the actual topics of the document are generated. The structure of a Hidden Markov Model we build is shown in Fig.4.

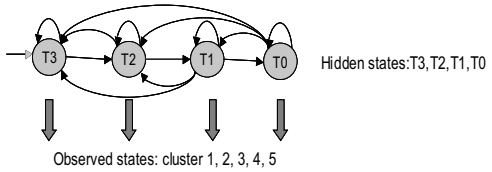

Figure 4. The structure of a Hidden Markov Model. In this case , the number of states is 4.

- Hidden states: $S = \{T3, T2, T1, T0\}$
  – Reaching a target page by 3 or more, 2, 1, 0 hop(s).
- Visible states: $K = \{1, 2, 3, 4, 5\}$
  – Cluster number which web pages belong to.
- HMM parameters θ:
  – Initial Probability Distribution Matrix $\pi = \{P(T_0), P(T_1), P(T_2), P(T_3)\}$.
  Probability of reaching a target by 0, 1, 2, 3 or more hop(s) at time 1, respectively.

– Matrix of Transition Probabilities: $A = [a_{ij}]_{4\times 4}$, where, a_{ij} = probability of being in the T_j state at time *t+1* given that you are in state T_i at time *t*. Fig.4 shows all possible transitions between all hidden states. Some state transitions are not possible. For example, there is no $T_3 \rightarrow T_1$, since if you can go from T_3 to T_1 in one hop, then the page is not a true T_3, it is a T_2. The same as $T_3 \rightarrow T_0, T_2 \rightarrow T_0$ transitions. It is possible to have $T_j \rightarrow T_j (j = 0..3)$ transitions. This will happen if there are cross links at the same distance from the targets.

– Matrix of Emission Probabilities: $B = [b_{ij}]_{4\times 5}$, where b_{ij} = probability of seeing cluster *j* if you are in state T_i.

To estimate the HMM parameters θ, we "label" all nodes in the concept graph as T_0, T_1, T_2, T_3 in a Breadth-First search out of the set of target pages (T_0) as shown in Fig. 5.

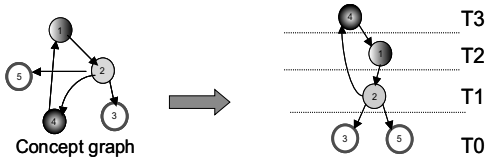

Figure 5. Parameter Estimation of HMM. Red-circled nodes are user marked target pages.

2.4 Focused Crawling

The crawler utilizes a queue, which is initialized with the starting URL of the crawl, to keep all candidate URLs ordered by their visit priority value. We use a timeout of 10 seconds for web downloads and filter out all but pages with text/html content. The crawler downloads the page pointed to by the top URL of the queue, calculates its reduced LSI space representation introduced in 2.2, and extracts all the outlinks. Then all children pages will be downloaded and classified using the K-Nearest Neighbor algorithm into a cluster to obtain the corresponding visit priority value based on the learned HMM.The value of K is chosen as the size of smallest cluster.

The resulting probability $P(X \mid \theta)$ for HMM θ learned from section 2.3 with a given observation web page sequence X is calculated with the Viterbi algorithm, which delivers the probability of the most likely hidden state sequence for the HMM θ. In particular, we define the partial probability δ, which is the probability of reaching a particular intermediate hidden state. For each crawled page *t*, we calculate all partial best paths for each hidden state, each of them has an associated probability δ. We find the overall best path by choosing the state i with the maximum partial probability $\delta(i, t)$, which is the maximum probability of all partial paths ending at state i at page t. As a result, a page whose cluster number has higher estimated probability of reaching to the targets will be assigned a higher priority value. We expect that URLs on the top of the queue will locate targets rapidly.

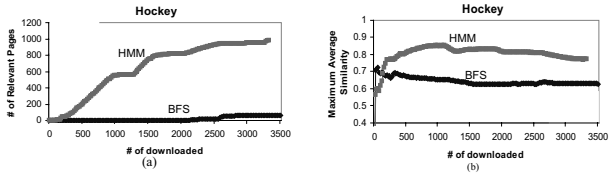

Figure 6. Topic *Hockey:* (a) # of relevant pages retrieved by Best-First crawler and our system using HMM. (b) Maximum Average Similarity of all downloaded pages by Best-First crawler and our system using HMM.

We also set a relevant threshold for determining if a web page is relevant to the user's interests. If its maximal cosine similarity to the target set is greater than the threshold, the URL will be stored and presented to the user as a relevant page.

3 Experiments

To evaluate the effectiveness of our approach, we use precision measure, which is the *percentage* of the web pages crawled that are relevant. Because there are multiple target pages, the relevance assessment of a page p is based on maximal cosine similarity to the set of target pages T with a confidence threshold γ. That is, if $\max_{t\in T} \cos(p, t) \geq \gamma, p$ is considered relevant.

We also calculate the *Maximum Average Similarity* σ. That is,

$$\sigma = \max_{t\in T} \frac{\Sigma_{p\epsilon S} \cos(p, t)}{|S|} \quad (1)$$

where T is the set of target pages found, S is the set of pages crawled, $\cos(p, t)$ is the standard cosine similarity function.

3.1 Results

We selected a variety of topics which are neither too broad nor too narrow, as shown in Table 1. For comparison, we chose Best-First Search (BFS) crawler. Best-First crawling outperforms other focused crawling strategies according to [6]. Its visit priority order of URL is based on maximal cosine similarity between the set of target pages and the page where link was found.

Fig.6(a) shows the number of relevant pages against the number of pages downloaded for the topic *Hockey*. The results show the significant improvement over Best-First crawling. Obviously, Best-First crawling pursued the links that appear the most promising at the expense of longer term huge loss, whereas our focused crawler explored the suboptimal links that eventually lead to longer term and larger gains, in spite of penalty at the early stage of the crawl, as shown in Fig.6(b). Fig.7(a) shows another topic *Linux*

Table 1. Different kind of crawls

Topics	# of pages user visited	# of targets user marked	γ	Start URL
/Sports/hockey	207	34	0.7	http://sportsnetwork.com, http://about.com/sports
/Computers/Software/Operating _Systems/Linux	200	30	0.9	http://www.computerhope.com,http://about.com/compute http://comptechdoc.org/os, http://www.informationweek.com/techcenters/sw/
/Health/Condition_Diseases/Diabetes, Heart Diseases	112	18	0.8	http://www.cnn.com
/Health/Condition_Diseases/Diabetes, Heart Diseases	112	18	0.8	http://www.nih.gov/, http://www.healthfinder.gov/ http://www.healthatoz.com/, http://www.healthweb.org/

for which our system showed the least average performance improvement over Best-First crawler, although it still outperforms Best-First crawler. We found that such topics are those with long paths and large subgraphs where topical coherence persists, so Best-First crawler performs well too. However, our system locates relevant pages quickly at the beginning. Note that the early stage is very important for focused crawling in cases of searching for dynamic content which changes quickly such as news, time-sensitive materials and personalized information.

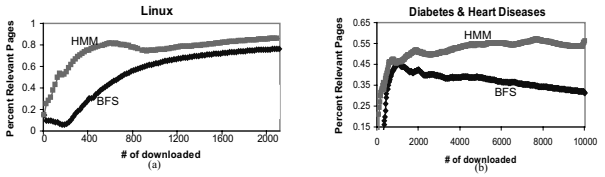

Figure 7. Precision measure by Best-First crawler and our system using HMM on different topics.

Another advantage of our system is focusing on what the user really wants on specific topics. User's interests are not limited to one topic and single domain. Fig.7(b) shows the results of crawling topics *Diabetes* and *Heart Disease* for the user at the same time and the numbers of relevant pages retrieved by our system are 50% more than those by Best-First crawler when crawling 10,000 pages. Fig.8 shows the results of the same topic but different start URL.

4 Conclusion and Discussion

Our focused crawler system uses HMM to model the link structure and content of documents leading to target pages by learning from user's topic-specific browsing. Our system is unique in several respects.The proposed way of capturing and exploiting user's personalized interests can potentially lead to focused crawlers that retrieve the information which is the most relevant to the user's interests. The prediction is based on hidden semantic content and linkage hierarchy leading to targets instead of only physical link distance. We show that our system locates relevant pages quickly and is superior to standard focused crawler. In ongoing work, we continue to conduct empirical studies for of the effect of clustering algorithms and we extend feature space by adding anchor texts, titles, keywords and user feedback.

Our user learning model is versatile, and potentially suitable for different applications. The system does not rely on

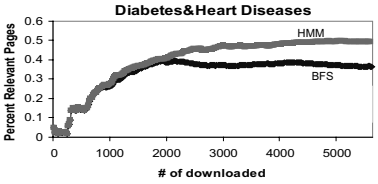

Figure 8. Topic is *Diabetes and Heart Disease* and start URL is *www.cnn.com*

other search engines and the parameters can be tailored to different users and topics. It can be used to build topic-specific portals, while individual users can rely on it for personal online search.

References

[1] C. Aggarwal, F. Al-Garawi, and P. Yu. Intelligent Crawling on the World Wide Web with Arbitrary Predicates. In *Proceedings of the 10th International WWW Conference*, Hong Kong, May 2001.

[2] S. Chakrabarti, M. van den Berg, and B. Dom. Focused Crawling: a new approach to topic-specific web resource discovery. In *Proceedings of the Eighth International WWW Conference*, Toronto, Canada,1999.

[3] M. Diligenti, F. Coetzee, S. Lawrence, C. Giles, and M. Gori. Focused Crawling Using Context Graphs. In *Proceedings of the 26th International Conference on Very Large Databases (VLDB 2000)*, Cairo, Egypt, September 2000.

[4] P. Lyman, H. Varian, J. Dunn, A. Strygin, and K. Swearingen. How much information? 2003. *http://www.sims.berkeley.edu/research/projects/how-much-info-2003/*, School of Information Management and Systems,Univ. of California at Berkeley, Accessed Mar. 2004.

[5] F. Menczer and R. Belew. Adaptive retrieval agents: Internalizing local context and scaling up to the web. *Machine Learning*, 39(2/3):203–242, 2000.

[6] F. Menczer, G. Pant, P. Srinivasan, and M. Ruiz. Evaluating Topic-Driven Web Crawlers. In *Proceedings of the 24th Annual International ACM/SIGIR Conference*, New Orleans, USA, 2001.

[7] J. Rennie and A. McCallum. Using Reinforcement Learning to Spider the Web Efficiently. In *Proceedings of the Sixteenth International Conference on Machine Learning(ICML-99)*, pages 335–343, 1999.

[8] S.C.Deerwester, S.T.Dumais, T.K.Landauer, G.W.Furnas, and R.A.Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.