A Novel Approach for Semantic Prefetching Using Semantic Information and Semantic Association

Sonia Setia, Jyoti and Neelam Duhan

Abstract Exponential growth of web accesses on the Internet causes substantial delays in providing services to the user. Web prefetching is an effective solution that can improve the performance of the web by reducing the latency perceived by the user. Content on the web page also provides meaningful data to predict the future requests. This paper presents a content-based semantic prefetching approach. The proposed approach basically works on the semantic preferences of the tokens present in the anchor text associated with the URLs. To make more accurate predictions, it also uses the semantic information which is explicitly embedded with each link. It then computes the semantic association between the tokens and links then associates weightage in order to improve the prediction accuracy. This prefetching scheme would be more effective for long browsing sessions and will achieve good hit rate.

Keywords Web prefetching · Content-based prefetching · Semantic prefetching · Semantic information · Semantic association · Prediction

1 Introduction

Today, the number of users accessing the web has increased tremendously, which resulted into the increased traffic in the network as well as load on the web servers. Users are demanding the techniques which can reduce the latencies perceived

S. Setia (✉) · Jyoti · N. Duhan
Department of Computer Engineering, YMCA University of Science and Technology, Faridabad, India
e-mail: setiasonia53@gmail.com

Jyoti
e-mail: justjyoti.verma@gmail.com

N. Duhan
e-mail: neelam_duhan@rediffmail.com

V.B. Aggarwal et al. (eds.), *Big Data Analytics*, Advances in Intelligent Systems and Computing 654, https://doi.org/10.1007/978-981-10-6620-7_45

during web access. To reduce this latency, many techniques have been developed in last few years. One of them is prefetching.

Prefetching is a technique which pro-actively fetches the web pages before the user explicitly demands those pages. To predict the web pages which are likely to be accessed next, content-based prefetching works on the semantic preferences of the pages retrieved in the past. It is found that user surfing is always done by using the anchor texts of URLs. The anchor text provides the description of the links. For example, a user having interest in shopping of a particular brand say 'AMUL' would like to see all the products of 'AMUL.' This is the phenomena of 'Semantic Locality.' This paper works on the similar concept as that of content-based semantic prefetching.

The remainder of this paper is organized as follows: Sect. 2 presents the related work for web prefetching techniques. Section 3 describes the proposed framework. Section 4 discusses the process of proposed technique. Finally, Sect. 5 concludes this paper.

2 Related Work

Many Prefetching techniques are provided in literature. Khan et al. [1] gave a new idea of content-based prefetching. Authors stated that prediction algorithm can give more accurate results if it considers the web space organization. A new prefetching technique is suggested in this paper where predictions are made by analyzing the contents of the current web page. Each web page contains a number of links on it.

Ibrahim et al. [2, 3] introduced a keyword-based semantic prefetching technique where predictions are made based on the semantic preferences of past retrieved web documents. This technique was applied to the Internet news services. This paper motivated the fact that user's surfing is guided by the keywords present in URL anchor text.

Venketesh et al. [4] used the tokens, generated from anchor texts associated with the URLs on the web page, to make effective predictions. Authors applied Naïve Bayes Classifier to compute the probability values of each URL, based on which the preference list is generated to make prefetching more effective.

Sharma and Dubey [5] proposed a semantic-based prefetching scheme which uses decision tree induction to compute the probability of each link to be clicked next. SPRINT-Decision tree induction method [6] is used in the proposed framework. Pruning is also applied on the decision tree so as to improve performance.

Setia et al. [7] provided a review in the area of web prefetching. The paper basically focuses on the various web prefetching techniques. Eight categories of web Prefetching were reviewed with detailed study of each.

Hu et al. [8] proposed a method to organize the multimedia resources on the web. Authors used the semantic link network model to find the associations between multimedia resources. In this paper, social tags and the surrounding text associated with the multimedia resources are used to determine the semantic

association between them. The proposed model provides a new method to organize the multimedia resources with their semantics.

There are so many content-based Prefetching techniques in the literature which are using keywords present in the anchor texts of web objects. Existing techniques consider that user surfing is always done by using the anchor texts of URLs where anchor text provides the description of the links. For example, a user having interest in shopping of a particular brand say 'AMUL' would like to see all the products of 'AMUL.' This is the phenomena of 'Semantic Locality.' But these techniques don't consider the semantics of the keywords associated with anchor texts of links as different anchor texts may be used by the web page designer to describe the page. For example, one designer can use 'apple' and other can use 'iphone' for the mobile phone manufactured by 'Apple.' Although these two are different keywords, but if user is interested in the mobile phone manufactured by 'Apple' he would like to see both the pages described by either 'Apple' or 'iphone.' Thus, these two keywords are semantically related to each other. Therefore, it would be more efficient to compute the semantic association between two tokens for better prediction.

3 Proposed Framework for Semantic-Based Prefetching

Proposed framework provides an efficient approach for more accurate predictions of the web pages by resolving the issues discussed above. Proposed approach consists of following components (Fig. 1):

- Semantic Link Analyzer, Token Extractor, Prediction System, Prefetching System, Storage Cache.

A. Semantic Link Analyzer

Semantic Link Analyzer analyzes the links on the web page and determines the priority order of links which should be considered first for evaluation. For this,

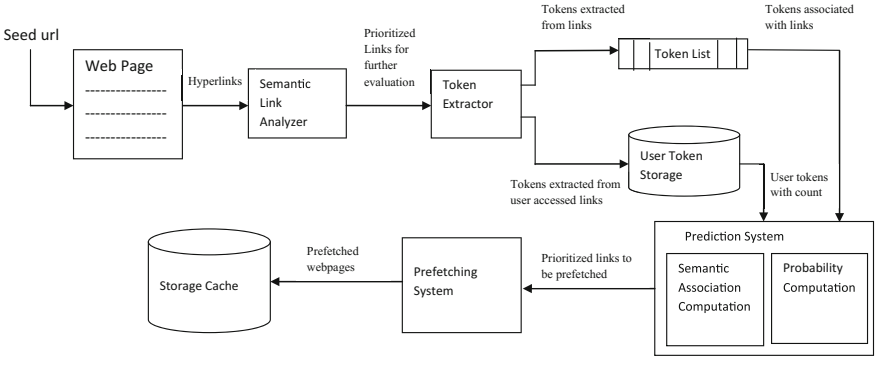

Fig. 1 Components of the system

Semantic Link Analyzer uses the semantic information on a web page that is added while designing the web page i.e., a semantic type is associated with each hyperlink using XML tags. Semantic type reflects a semantic relation between two web pages. Semantic types are defined as follows [9]:

1. Sequential (seq): This type indicates that these two pages should be accessed in a sequence i.e., one after another.
2. Similar (sim): This type indicates that both pages are semantically similar.
3. Cause-Effective (ce): This type indicates that one page is the cause of another.
4. Implication (imp): This type indicates that the semantics of one page implies the other.
5. Subtype (st): This type indicates that one is a part of another.
6. Instance (ins): This indicates that one is an instance of other.
7. Reference (ref): This indicates that one page is a detailed explanation of the other page.

The relative semantic strength orders of these types are as follows:
ref < ins < st < imp < ce < sim < seq

When the user requests for a web page server sends the page with the semantic information associated with each hyperlink on that page. Semantic Link Analyzer extracts the URLs from that page and analyzes all of those links and prioritizes them according to their semantic strength. If more than one link are of same type then relative location of the link on that page is considered for prioritization. This ordered list of links is used by the token extractor for further evaluation.

B. Token Extractor
This component takes the prioritized links as input from the Semantic Link Analyzer. As a large number of links may be there on a web page this can't be examined at a time. Therefore, a fixed '*n*' number of links are considered for further examination. Thus, it takes the first set of links i.e., first '*n*' number of links from the prioritized list of links and extracts the anchor text associated with these links. Further anchor texts are processed to generate the set of tokens. This component maintains two data structures in turn to store the information.

(i) Token List
Token List contains the tokens extracted from the anchor text associated with the URLs on the current page. Based on these tokens, probability of each URL is computed corresponding to the token count is maintained in User Token Storage.

(ii) User Token Storage (UTS)
Whenever user accesses a web page, the tokens generated from the anchor text associated with that page is added to the User Token Storage. Thus, this unit stores the information about the user's interest in a particular topic. With each token, count value is also maintained in User Token Storage. Token count reflects the number of times a token appears in the

anchor text associated with the user's requested page. Whenever the token appears in the requested page, its associated count gets incremented by one if it already exists in the storage unit otherwise new entry is created with count value one. These tokens contained in storage unit are used by the prediction system to compute the probability of URLs being accessed in near future.

C. Prediction System
Prediction System is responsible for making predictions of the future web pages. This is being done based on two computations:

(1) Semantic Association Computation [8]

Different anchor texts may be used by the web page designer to describe the page. For example, one designer can use 'apple' and other can use 'iPhone' for the mobile phone manufactured by Apple. Although these two are different keywords, but if user is interested in the mobile phone manufactured by 'Apple' he would like to see both the pages described by either apple or iPhone. Thus, these two keywords are semantically related to each other. Therefore, it would be more efficient to compute the semantic association between token set of a link and each token present in User Token Storage. If any token is found semantically related to the token set of a particular link, then that token will be included in the token set of that particular link. This would give the more weightage to that particular link and will also help in computing the more accurate probability of that link to be accessed in future.

Semantic association is computed between two token sets using following steps:

- Let 'N' be the number of entries in the token list corresponding to N links on the current page, and each entry is a token set $T_i = \{t_1, t_2 \ldots t_m\}$ extracted from the anchor text of a link and $1 \leq m \leq n$; 'n' is a positive integer and $1 \leq i \leq N$.
- Let 'M' be the number of entries in User Token Storage, and S_j represents a token in User Token Storage and $1 \leq j \leq M$.
- $\mathrm{SA}(T_i, S_j)$ is the semantic association between two token sets and it is computed as

$$\mathrm{SA}(T_i, S_j) = \frac{\sum_{t_k T_i,\ s_l S_j} \mathrm{SA}(t_k, s_l)}{|T_i|} \tag{1}$$

where $\mathrm{SA}(t_k, s_l)$ is the semantic association between two tokens and it is computed as follows:

$$\mathrm{SA}(t_k, s_l) = \log\left(\frac{M * M(t_k \cap s_l)}{M(t_k) * M(s_l)}\right) / \log M \tag{2}$$

where

- M is the number of web pages in the search engine.
- $M(t_k)$ denotes the page counts for the token t_k.
- $M(s_l)$ denotes the page counts for the token s_l.
- $M(t_k \cap s_l)$ denotes the page counts for the query $t_k \cap s_l$ which measures the co-occurrence of the tokens t_k and s_l.

If this value is greater than a fixed predefined threshold then tokens will be considered semantically related to that token set and that particular token will be included in the token set associated with a link i.e.

$$T_i = T_i \cup S_j$$

For that particular link, this token set will be considered for further computation.

(2) Probability Computation

Finally, the system computes the probability of each link using Naïve Bayes classifier. Set of tokens associated with a particular link is taken, and the count value corresponding to these tokens is compared to the total tokens count in User Token Storage to determine its probability to be clicked next.

Probability of appearance of a link for a given storage S

$P(T_i/S)$ is computed by taking the product of individual tokens probabilities:

$$P(T_i/S) = \prod_{i=1}^{m} [C + P(t_k/S)] \tag{3}$$

where

- $P(T_i)$ = Probability of appearance of a link
- $P(S)$ = Probability of User Token Storage
- $P(t_k)$ = Probability of individual token associated with a link and $t_k \in T_i$
- $P(t_k/S)$ = Probability of each token for a given storage S which is computed as

$$P(t_k/S) = \frac{\text{Count of } t_k \text{ in } S}{\text{Total count of tokens in } S}$$

- C is a Constant with value '1' which is added to each token probability whether it is present in User Token Storage or not. It is added to avoid two cases:

1. Probability of link to be less than individual token probability
2. To avoid zero probability situation because product value becomes zero if few tokens of a link are not present in User token Storage

Based on these probabilities of links, prediction system generates a priority list of links needed to be prefetched and top priority is given to the links having high probability.

D. Prefetch System

Prefetch System takes the priority list generated by prediction system as input and prefetches the corresponding web pages from the server and stores them in storage cache. When user clicks a link, then any ongoing prefetching will be suspended and system will look for the requested page.

E. Storage Cache

Web pages that are prefetched by the prefetch system are stored in storage cache to satisfy the future requests made by the user. Since the cache is of limited size, replacement of web page would be required when cache gets full. In this paper, Least Recently Used (LRU) algorithm is used as cache replacement algorithm. It removes the web pages from the cache that are not accessed for a long time and makes sufficient space for new pages.

4 Process of Prefetching

The whole process (Fig. 2) involves the following steps:

1. User opens a browser and requests for a page by entering the URL.
2. Server sends the corresponding web page including the semantic information associated with each link on that page which is displayed to the user.
3. Semantic Link Analyzer extracts the links and semantic information associated with each link present in the page and gives a prioritized list of links based on their relative strength and position on the page.
4. Out of these links, first '*n*' set of links are considered for further evaluation.
5. Anchor text is extracted associated with each link. These are further processed to generate the set of tokens where each token refers to single word.
6. These tokens are stored in the Token List.
7. Whenever user clicks a link, tokens generated from the anchor text associated with that link are being added to the User Token Storage with count value '1' if it doesn't exist in storage unit otherwise, count value gets incremented by one each time token appears in user access.
8. Computing the semantic association between the anchor text associated with each link presents in token list and the tokens present in User Token Storage. If it is found greater than the threshold value (in this paper, it is manually computed) then that token is semantically related to the anchor text of a link and it will be added to the token set of the anchor text of that particular link to compute its probability to be accessed in future.

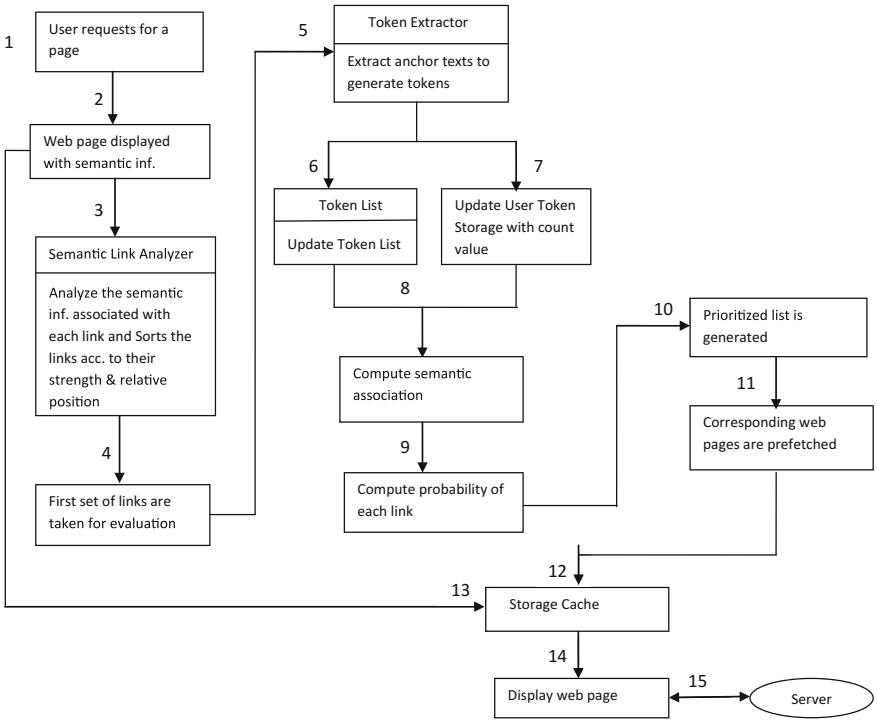

Fig. 2 Process of prefetching

9. Compute the probability of each link to be accessed next using Naïve Bayes classifier.
10. Based on these probability values, priority list of links will be generated.
11. Prefetching will be done based on the priority list.
12. Prefetched web pages are stored in storage cache which is being managed using LRU replacement algorithm.
13. Whenever user clicks a link on the web page, ongoing prefetching will be suspended and system looks for that page in storage cache.
14. If it is present in cache, it is displayed to the user without any delay.
15. Otherwise, it is retrieved from the server and displayed to the user.

5 Conclusion

This paper has presented a novel approach based on semantic prefetching which uses the anchor texts associated with the URLs present on the current page for making effective predictions. Besides this, it also uses the semantic information which is explicitly embedded with each link, to prioritize the links at the first step. This approach basically works on the semantic preferences of the tokens present in the anchor text associated with the URLs. To compute the accurate probability of each link to be prefetched, this approach computes the semantic association between the anchor text of the URLs and tokens present in user token storage so that more accurate weightage can be given to each link if it is found semantically related to any token. The proposed approach helps to minimize the user perceived latency by making more accurate predictions and achieving maximum hits.

References

1. Khan, J.I., Tao, Q.: Exploiting Webspace organization for accelerating Web prefetching. In: Proceedings of the IEEE/WIC International Conference on Web Intelligence, Halifax, Canada (2003)
2. Ibrahim, T.I., Xu, C.: Neural net based predictive prefetching to tolerate WWW latency. In: Proceedings of the 20th International Conference on Distributed Computing Systems (2000)
3. Xu, C.Z., Ibrahim, T.I.: A keyword-based semantic prefetching approach in Internet news services. IEEE Trans. Knowl. Data Eng. **16**(5), 601–611 (2004)
4. Venketesh, P., Venkatesan, R., Arunprakash, L.: Semantic Web prefetching scheme using Naïve Bayes classifier. Int. J. Comput. Sci. Appl. **7**(1), 66–78 (2010)
5. Sharma, N., Dubey, S.K.: Semantic based Web prefetching using decision tree induction. In: Proceedings of 5th International Conference on the Next Generation Information Technology Summit (2014)
6. Shafer, J., Agrawal, R., Mehta, M.: SPRINT- a scalable parallel classifier for data mining. In: Proceedings of 22nd International Conference on Very Large Database, pp. 544–555 (1996)
7. Setia, S., Jyoti, Duhan, N.: Survey of recent Web prefetching techniques. Int. J. Res. Comput. Commun. Technol. **2**(12) (2013)
8. Hu, C., Xu, Z., Liu, Y., Mi, L., Chen L., and Luo, X.: Semantic link network- based model for organizing multimedia big data. IEEE Trans. Emerg. Top. Comput. **2**(3) 2014
9. Pons, A.P.: Object prefetching using semantic links. Database Adv. Inf. Syst. **37**(1) (2006)
10. Venketesh, P., Venkatesan, R.: Adaptive Web prefetching scheme using link anchor information. Int. J. Appl. Inf. Syst. **2**(1) (2012)