Analysis of the Library Novelty Search Service Scheme Based on ICT Technology

Siyu Xia(✉)

Library, Wuhan University of Technology, No.122 Luoshi Road, Wuhan, China
siyu.xia@whut.edu.cn

Abstract. With the rapid developments of fifth-generation (5G) mobile networks and Cloud computing, an increasing number of advanced information and communication technology (ICT) technologies are coming into people's daily life. As one of the most critical sections of smart library, the research of the library novelty search service has attracted significant attention in the last few years. In this paper, an intelligent libraries novelty search service model is proposed. To improve the novelty search service quality, NSAs' novelty service pre-allocation and NSAs' novelty search assignment scheme is proposed. The results show that the proposed scheme is capable of improving the successful ratio of novelty search service requests.

Keywords: Smart Library · Novelty Search Service · Digital Library

1 Introduction

In the fifth-generation (5G) mobile networks and Cloud computing era, digital information can be efficiently distributed via a collection of ways such as social media platforms, smartphone apps, newspapers, hardcopy books, journals and so on [6, 13]. In the past, the traditional library was generally regarded as an information platform, constructed with a specific information target that functions as an intermediary between different databases and readers. However, the role of the library has been transforming from passive manner Web 1.0 to active manner Web 2.0 onward, where one of the most important features of Web 2.0 is user participation, bring excellent user experience, dynamic content, immersive experience to readers. The reference [8] reported that the concept of Library 2.0 could be specified as a subset of traditional library services to satisfy users' different service requests precipitated in conjunction with Web 2.0. Since Web 2.0 was developed into Web3.0, which combines artificial intelligence with Web 2.0 applications, it brings opportunities for the use of semantic tagging and annotation for the social web. The intelligence of Web 3.0 has been inspired the development of Library 3.0, where we are already reached. The concept of Web 4.0 is widely discussed as an intelligent agent that is integrated with advanced human interaction interfaces. Some researchers indicated that Web 4.0 might bring new opportunities for Library 4.0, which include software-based user-friendly interfaces and low-latency service latency to promise service quality [13, 16, 18].

Z. Zhan et al. (Eds.): ICBDIE 2022, AHCS 5, pp. 460–469, 2023.
https://doi.org/10.2991/978-94-6463-034-3_47

Technological enhancements have improved higher education to unprecedented altitudes, where a number of outstanding universities and institutions worldwide are becoming highly likely to upgrade their facilities and services, especially in the perspective of an innovative campus [5]. Universities that deploy intelligent digital infrastructure and provides services to their students and staff, with the aid of the connectivity of intelligent devices enabled by the Internet of Things (IoT), information and communication technology (ICT) and even big data, bring revolutionary benefits to the campus, which is a promising way to realize the smart campus [3, 11, 14, 17].

As one of the most important research topics in smart campus, the smart library has been received considerable attenuation from both academia and industry. In the last few years, the concept of the smart library has been widely investigated as a significant component to realize the smart campus [4]. The definition of the smart library can be stated as an interactive, innovative, informative and international way to provide library services to readers either in virtual or real. The smart library can be regarded as an integration of intelligent hardware and software with plenty of chances for searching and providing digital information to readers based on their inquiries and service requirements. As one of the essential parts of libraries' daily work, advanced services such as novelty search, hypertext, extensive data services, and personalized services, should be considered and developed with numerous up-to-date ICT methods, such as IoT, Radio Frequency Identification, Cloud computing, smart sensors and so forth.

Due to the rapid developments of library novelty search services, a new technical trend is to deploy cloud-like resources to the network edge, such as deploy caching serves integrated with a library information center. This brings opportunities for smart libraries to go beyond in the 5G era, transforming traditional libraries from just library service providers to multiple novel services providers [9]. For example, as one of the most significant services for the smart library, scientific and technological novelty search service demands vast human resources and suffer time cost considerably [10, 12]. However, few works jointly consider the advanced ICT methods and scientific and technical novelty search services [1, 2, 15].

This paper jointly considers Cloud computing and novelty search services. First, the smart library novelty search model is proposed. Moreover, with the aim to balance novelty service applicants (NSAs) demands and novelty service providers' (NSPs) service quality, the optimization problem is formulated and analyzed. To solve the challenging formulated problem, a heuristic algorithm is proposed. Finally, the selected significant performance of the proposed scheme is listed and compared with some advanced technical methods proposed in the latest references.

2 System Model and Problem Formulation

The proposed smart library novelty search service model is demonstrated in Fig. 1, where a set of $\mathcal{N}$ submitted novelty services from NSAs, a set of $\mathcal{M}$ NSPs, a set of $\mathcal{K}$ caching servers and the corresponding novelty reports. In this system, caching servers are integrated with the library information center. After each NSP finish the novelty search service, the system will assign caching disk to store the novelty report. Moreover, the content in caching servers can be adjustive updated by NSPs based on their historical

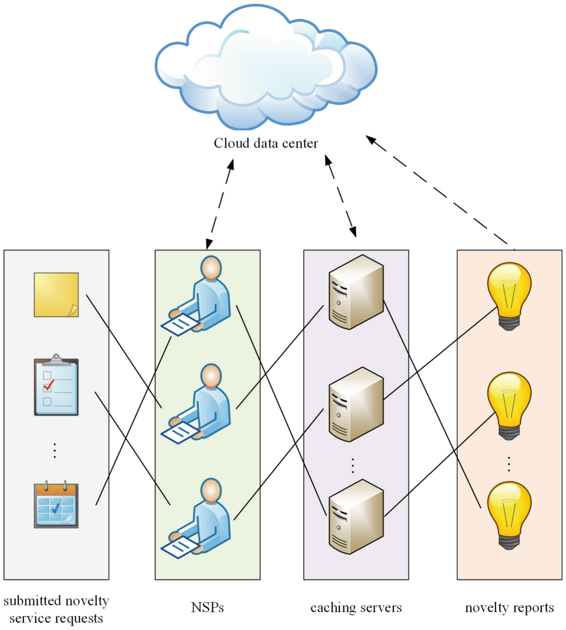

Fig. 1. The proposed intelligent libraries novelty search service model.

data. One should note that one novelty service request can only be processed by one MSP. The data transmission latency between caching servers and Cloud data center is ignored in comparison with the report processing time cost. The derivations proposed in this paper can be generalized and extended to Cloud-based smart library novelty search architecture.

2.1 NSAs' Novelty Search Service Model

At the beginning of each equal-length time slot, each NSA may have different QoS preferences novelty search requests, including pricing, time cost and so forth. The set of novelty service requests can be mathematically formulated as the set denoted by $\mathcal{N} = \{1, 2, \ldots, N\}$. Each element $i \in \mathcal{N}$ can be characterized by numerous key components, denoted by submitted data size D_i, service complexity W_i and the time cost allowance T_i. In this respect, the novelty service can be formulated as

$$U_i = (D_i, W_i, T_i) \tag{1}$$

Note that each novelty report is required to be finished not more than T_i. Each novelty report should be uploaded to the Cloud data center for data analysis and storage. Afterwards, Cloud data center may generate service quality report to guide NSPs and library information center to improve their work efficiency.

2.2 NSPs' Service Model

In this paper, one can assume that the workload and the service quality of NSPs may be different from each other. That is to say, different NSPs may have different service quality such as time cost, availability and so forth. It is evident that the number of currently processing novelty services will significantly slow down NSPs' processing speed to finish other novelty reports. Let q_j be the number of currently running novelty search services of NSP i. Define γ_j as the maximum number of novelty search requests that NSP i can handle simultaneously. One can observe that no new novelty search request can be assigned to NSP i if and only when the number of currently running novelty search services reach $\gamma_j = q_j$. One should also note that when q_j is approaching γ_j, the work efficiency of NSP i may become lower and the new income novelty search request may be better to assign to other NSPs with a smaller number of currently running novelty search services. Define s_j is the novelty search report processing speed when NSP j is fully occupied.

Since the number of the current running of each NSP j cannot exceed the maximum number of novelty search requests that can handle simultaneously, one can obtain that

$$q_j \le \gamma_j \tag{2}$$

To investigate the relationship between the current workload and the processing speed, one can follow the following equation as proposed in the number of recent research work to measure NSP i's processing speed, considering γ_j, q_j and β_j, one has [7]

$$f_j = \beta_j^{\gamma_j - q_j} \tag{3}$$

Assume that NSP j is assigned to serve NSA i, the normalization time cost to obtain the novelty report can be expressed as

$$T_{ij} = \frac{D_i W_i}{f_j} \tag{4}$$

By substituting Eq. (3) into Eq. (4), one can obtain the relationship between NSP i's processing speed and the submitted novelty service requirements. One has

$$T_{ij} = \frac{D_i W_i}{\beta_j^{\gamma_j - q_j}} \tag{5}$$

Some hot data, such as ESI highly cited references, may repeatedly searched or analyzed by NSPs and Cloud data center. In addition, some NSAs may frequently upgrade their outcomes, requesting new novelty search requests, where the majority of the novelty reports is highly similar with cached data in caching servers and Cloud data center.

In this paper, to investigate the similar content of the new novelty search requests and the cached data, one can assume the utilization of the cached data by NSPs is assumed to follow Zipf distribution. In this regard, the utilization of the cached data that can be used in the new novelty service request can be expressed as

$$P_i^k = \frac{\frac{1}{k^{\epsilon}}}{\sum_{k=1}^{K} 1/k^{\epsilon}} \tag{6}$$

where $\in$ is the shape factor of the Zipf distribution. Let the caching utilization indicator of novelty service i be the $\alpha_i^k \in \{0, 1\}$, where $\alpha_i^k = 1$ means some content of novelty service i can be obtained from caching servers or Cloud data center and $\alpha_i^k = 0$ otherwise. When $\alpha_i^k = 0$, the assigned NSP may suffer additional time cost to handle the novelty service since the value of W_i may be higher.

When $\alpha_{ij}^k = 1$, some content can be handled by caching servers. For example, when NSA *i's* novelty search service is allocated to NSP j, if some content can be reused from caching server k, the processing time cost is defined as T_{ij}^k. In this way, the time cost constraint should follow

$$\max\left\{T_{ij}^k, T_{ij}\right\} \le T_i \tag{7}$$

Equation (7) means that each novelty search service should be finished within time allowance T_i. Since each report can only be handled by one NSP or obtained from caching servers.

2.3 Problem Formulation

In this paper, the novelty search service quality is defined to reflect the relationship between NSPs' work efficiency and novelty service time cost. Define λ is the weighting factor to reflect the workload of novelty search request i that can handed by the assigned NSP j and cached data stored in caching server with $0 \le \lambda_i \le 1$. When $\lambda_i = 0$, it means the novelty search service i is totally finished by the NSP j and $\lambda_i = 1$ indicates that the novelty search service i is entirely finished by the cached data, that is to say, novelty search service i is repeatedly submitted by different NSAs. The aim of the library novelty search service scheme is to minimize each novelty search service time cost by jointly considering NSPs' processing speed, NSAs' novelty search service quality and caching servers, which can be formulated as

$$\mathcal{P}1\text{: } \min \lambda T_{ij} + (1-\lambda)\alpha_{ij}^k P_i^k \tag{8}$$

Subject to
(3), (7)

Remark: The proposed problem $\mathcal{P}1$ is challenging to be solved since the optimization variable α_i^k is an 0-1 variable. Moreover, different optimization variables are closely coupled. Furthermore, note that due to the existence of Eq. (2), each NSP j's processing speed would decrease with the increase of workload q_j, the classical matching algorithm, e.g., Hungary algorithm becomes inefficient to solve $\mathcal{P}1$. In addition, the existing optimization methods such as convex optimization, dynamic programming and so forth cannot directly be tackled $\mathcal{P}1$.

3 The Proposed Method to the Problem

3.1 NSAs' Novelty Service Pre-allocation Scheme

Owing to the limited availability and processing speed of NSPs, one effective way to reduce the time-latency of novelty service is to improve the utilization of cached data in caching servers. Moreover, since not all submitted novelty search requests can be implemented by cached data, some emergency novelty search requests with high service complexity W_i can be assigned to NSPs with currently running novelty search services. In this way, one can analyze the novelty service requirements to obtain the services that cannot be successfully handled. The detailed information regarding the proposed pre-allocation algorithm is summarized in Algorithm 1. First, when the new novelty service request i is submitted, the system will check all NSPs' availability and obtain the corresponding f_j. Second, if the submitted novelty service request i have the similarity content with the cached data in caching servers, one can assign $\alpha_{ij}^k = 0$; otherwise, the system will assign the NSP j with a higher value of f_j to handle the novelty service request i. Furthermore, if there exists the condition, e. g., $\max\left\{T_{jj}^k, T_{ij}\right\} > T_i$, it means that the novelty service request i cannot be handled anywhere due to the strict time cost constraint.

Algorithm 1: The proposed pre-allocation algorithm

Inputs: $U_i,\ f_j$
  If $\alpha_{ij}^k = 0$
    Assign NSP with higher f_j;
  End If
  If $\alpha_{ij}^k = 1$
    Assign NSP with lower f_j;
    Obtain data from caching disk k;
  End If
  Obtain T_{jj}^k and T_{ij};
  If $\max\{T_{jj}^k, T_{ij}\} \leq T_i$
    Novelty search service i is successfully handled;
  End If
  If $\max\{T_{jj}^k, T_{ij}\} > T_i$
    Novelty search service i is failed;
  End If
Update $\mathcal{N}$ to $\widetilde{\mathcal{N}}$.
Outputs: novelty search requests pre-allocation.

3.2 NSAs' Novelty Search Assignment Scheme

After obtaining the updated set of novelty services of NSAs $\tilde{\mathcal{N}}$, one can investigated the relationship between NSAs' novelty services and NSPs' service. At each time slot, novelty services with higher similarity content stored in caching servers are highly likely to be handled with less NSPs' workload. For example, if the submitted novelty service i is composed of two parts, namely data block one U_i^1 and data block two U_i^2, where the first part should be handled by NSP and the latter one can be obtained from cached data as mentioned in Algorithm 1. In this manner, one can only consider U_i^1 to guide the NSP selection. In particular, one can sort each novelty service request $i \in \tilde{\mathcal{N}}$ in ascending order. In this way, the potential successful novelty service requests can be assigned with priority. In other word, novelty service requests with strict time requirements will be handled first. At this stage, one can obtain the priority novelty search requests. Afterwards, if some content can be obtained from cached space, i.e., $\alpha_{ij}^{k} = 1$, one can determine the P_i^k. In this way, one can obtain the time cost to finish the novelty report as $\min\{\lambda T_{ij}, (1-\lambda)\alpha_{ij}^{k} P_i^k\}$. The novelty report can be regarded as successful if and only if when $\min\{\lambda T_{ij}, (1-\lambda)\alpha_{ij}^{k} P_i^k\} \le T_i$.

However, when $\min\{\lambda T_{ij}, (1-\lambda)\alpha_{ij}^{k} P_i^k\} > T_i$, it means although the portion of submitted novelty service can be handled by utilizing cached data, the time cost still cannot satisfy the time cost allowance T_i. To reduce the value of $\min\{\lambda T_{ij}, (1-\lambda)\alpha_{ij}^{k} P_i^k\}$, one can check the availability of NSPs and assign the highest value of f_j to the priority novelty search request. One should bear in mind that some novelty search requests may still be unsuccessful due to the strict of time cost requirement. To solve this condition, one can advise MSAs to amend their time cost requirements or negotiate with MSPs to see whether it is possible to assign multiple MSPs cooperatively on this novelty search service. As such, the value of the target function proposed in Eq. (8) can be obtained. The proposed novelty search assignment algorithm is summarized in Algorithm 2.

Algorithm 2: The Proposed Novelty Search Assignment Algorithm.

```
For U_i, i ∈ Ñ
Sort elements in ascending order of T_i; //priority
Calculate the corresponding time cost T_ij;
  If some content can be obtained from cached space;
    Determine P_i^k;
    Obtain data from caching server k;
    Obtain T_ij^k;
      Obtain min{λT_ij, (1 − λ)α_ij^k P_i^k};
  End If
  If min{λT_ij, (1 − λ)α_ij^k P_i^k} > T_i
      Check the availability of NSPs;
      Sort f_j in ascending order;
        Assign the highest f_j to U_i, i ∈ Ñ;
        Obtain the updated min{λT_ij, (1 − λ)α_ij^k P_i^k};
 End If
 End For
```

4 Performance Evaluation

In this section, the performance of the proposed scheme is taken into account by using MATLAB software. The system performance evaluation was utilizing MATLAB 2021 in the 64-bit professional operating system, six-core 2.6 GHz Intel i7 processor with 16 GB RAM. The evaluation performance is investigated and compared with the proposed algorithm without utilizing caching technique. Assume that the maximum number of

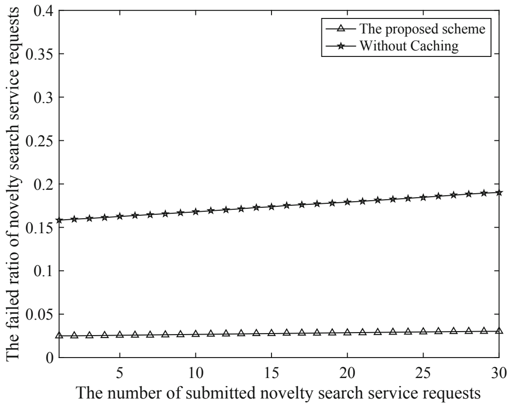

Fig. 2. The relationship between the failed ratio of novelty search service requests and the number submitted novelty search service requests under standard service complexity.

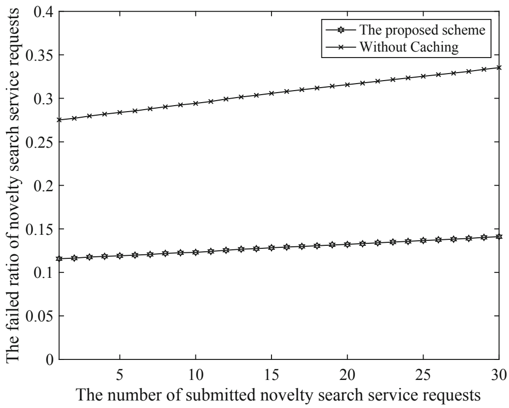

Fig. 3. The relationship between the failed ratio of novelty search service requests and the number submitted novelty search service requests under complex service complexity.

MSAs is 30, each MSA has a novelty search request and all requests are different from each other. There are 5 MSPs and the maximum number of novelty search requests that can handle simultaneously by one MSP is 6. The values of β_j and D_i are normalized for simplify purposes.

The relationship between the failed ratio of novelty search service requests and the number submitted novelty search service requests under simple and complex service complexity scenarios are given in Fig. 2 and Fig. 3, respectively. In Fig. 2, one can see that the failed ratio of novelty search service requests of the proposed scheme is around 4% when the number of submitted novelty search service requests is 15 while this value is about 17% without utilizing cached data. With the increase of the difficulty of the submitted novelty search service requests, e.g., higher value of W_i, the failed ratio of novelty search service requests of the proposed scheme is approaching 14% when the number of submitted novelty search service requests is 15 whereas the performance of without utilizing cached data is nearly 33%.

5 Conclusions

IN this paper, an intelligent library novelty search services scheme is proposed in the 5G and Cloud computing era. First, the problem of how to balance MSAs' novelty search requests and MSPs' working speed is investigated. Moreover, the proposed library novelty search service scheme is given. Finally, the performance of the proposed scheme is simulated and compared with the scheme without utilizing caching technology. The results show that the proposed scheme can significantly improve the failed ratio of novelty search service requests.

References

1. Aithal, P. S. (2016). Smart library model for future generations. International Journal of Engineering Research and Modern Education (IJERME), 1(1), 693-703.
2. Baryshev, R. A., Babina, O. I., Zakharov, P. A., Kazantseva, V. P., & Pikov, N. O. (2015). Electronic library: genesis, trends. From electronic library to smart library.
3. Cao, G., Liang, M., & Li, X. (2018). How to make the library smart? The conceptualization of the smart library. The Electronic Library.
4. Fang, S. (2020). Visualization of information retrieval in smart library based on virtual reality technology. Complexity, 2020.
5. Gul, S., & Bano, S. (2019). Smart libraries: an emerging and innovative technological habitat of 21st century. The Electronic Library.
6. He, D. (2020). A strategy of smart library construction in the future. Journal of Service Science and Management, 13(2), 330-335.
7. Li, T., Wang, K., Xu, K., Yang, K., Magurawalage, C. S., & Wang, H. (2019). Communication and computation cooperation in cloud radio access network with mobile edge computing. CCF Transactions on Networking, 2(1), 43-56.
8. Linaje, M., Preciado, J. C., & Sanchez-Figueroa, F. (2007). Engineering rich internet application user interfaces over legacy web models. IEEE internet computing, 11(6), 53-59.
9. Noh, Y., & Ro, J. Y. (2021). A study on the service provision direction of the National Library for Children and Young Adults in the 5G era. International Journal of Knowledge Content Development & Technology, 11(2), 77-105.
10. Ozeer, A., Sungkur, Y., & Nagowah, S. D. (2019, December). Turning a Traditional Library into a Smart Library. In 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE) (pp. 352–358). IEEE.
11. Pandey, J., Kazmi, S. I. A., Hayat, M. S., & Ahmed, I. (2017, December). A study on implementation of smart library systems using IoT. In 2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions)(ICTUS) (pp. 193–197). IEEE.
12. Schöpfel, J. (2018). Smart libraries. Infrastructures, 3(4), 43.
13. Shah, A., & Bano, R. (2020). Smart Library: Need of 21 st Century. Library of Progress-Library Science, Information Technology & Computer, 40(1).
14. Simović, A. (2018). A Big Data smart library recommender system for an educational institution. Library Hi Tech.

15. Wada, I. (2018). Cloud computing implementation in libraries: A synergy for library services optimization. International journal of library and Information Science, 10(2), 17-27.
16. Yu, K., Gong, R., Sun, L., & Jiang, C. (2019). The application of artificial intelligence in smart library. Advances in Economics, Business and Management Research, 100, 708-713.
17. Zhao, L. (2021). Personalized recommendation by using fused user preference to construct smart library. Internet Technology Letters, 4(3), e273.
18. Zimmerman, T., & Chang, H. C. (2018). Getting smarter: Definition, scope, and implications of smart libraries. In Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries (pp. 403–404).