Recommendation of Revisiting Web Pages Related to Currently Browsing Web Pages

Wenjia ZHANG†, Yoshiyuki SHOJI††, and Hiroaki OHSHIMA†

† Graduate School of Information Science, University of Hyogo
8–2–1 Gakuen-nishimachi, Nishi-ku, Kobe, Hyogo 651–2197, Japan
†† College of Science and Engineering, Aoyama Gakuin University
5–10–1 Fuchinobe, Chuo-ku, Sagamihara, Kanagawa 252–5258, Japan
E-mail: †ad22v040@guh.u-hyogo.ac.jp, ††shoji@it.aoyama.ac.jp, †††ohshima@ai.u-hyogo.ac.jp

Abstract A considerable amount of activities on the web involves revisiting web pages when processing repetitive tasks on computers. Browser support for revisiting mainly consists of bookmarks and browsing history, or focuses on recently visited pages. We propose a method for recommending previously visited pages that are informative for the task in progress. Revisiting pages may be different in the tracing process, but are eventually opened after starting a certain web activity in a temporal sense. The process of the proposed method consists of three parts. First, we collected spaced co-occurring web pages in the browsing history. Second, we constructed a system to output revisiting pages based on currently browsing web pages. Finally, we used this method to recommend revisited pages in several specific tasks.

Key words recommendation system, web browsing history, information seeking behaviour

1. Introduction

Computers are often used for processing a variety of tasks. Some pages are used repeatedly for the same task occurring every time. Each time these repetitive tasks are done, it is time consuming to find pages that users would like to visit again, and it can also be difficult or even impossible to find it. Either of the above is a waste of time and makes work less efficient. For example, a job-hunting student needs to write application forms for different companies, for each company, the application form is often not completed in one go and needs to be finished multiple times. When writing the application form for Company A, this student referred to other people's application forms on the job-related website. There is a number of application form examples for the same company on job-related websites, such as some popular companies with as many as 2,975 (as of this date, 6 January 2023). When the student wants to continue writing an application form for company A for the second time, the student has to search through 2975 examples browsing history to find the one wanted, either of which is time-consuming and may not be found. This study will be used for the web browsing history of the chrome browser on the computer to investigate recommendations for revisiting web pages that have a reference value to the web pages currently being viewed.

The aim of this study is to recommend pages to users that they can revisit based on the pages they are viewing. More than 45% of the pages that we visit on the web are pages that we have visited before [12]. So what kind of pages are considered necessary to revisit? Let's assume that a student tried to do a programming assignment, and searched for some coding grammar. He or she browsed dozens of pages including the following. She browsed the following pages.

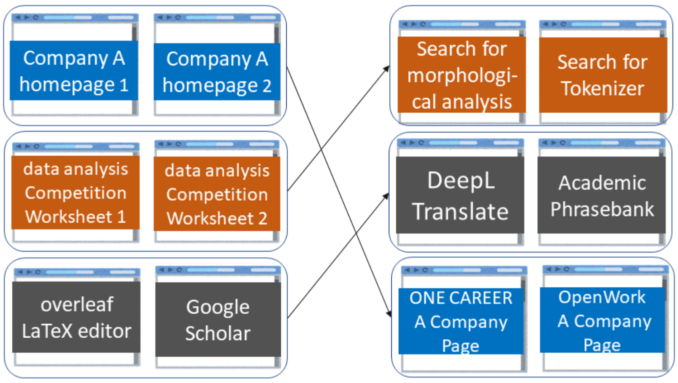

Figure 1 Example of recommendation

- The SERP for the query "pandas view rows specified column".
- The page for pandas: select rows/columns in dataFrame.
- The SERP for the query "Python recursive functions".
- The page for understanding the Python recursive func-

tions.

After the first task is finished, there is another task doing a machine learning-related assignment. She looked up some code for processing data and machine learning models and browsed the following pages.

- The SERP for the query "Least squares method Python".
- The page for Least Squares Regression in Python.
- The SERP for the query "machine learning dataset split".
- The page for Train-Test Split for Evaluating Machine Learning Algorithms.

In both tasks, she may have been writing code in the editor while referring to these pages. Although tasks are different and queries are not the same, there are still some connections between the two assignments. For example, there is

- Both are programming tasks.
- Both are related to "python" code.
- Pandas is available for splitting datasets.

In this study, the pages browsed in the programming task and the machine learning task is considered to be related. When doing the second assignment, the pages browsed in the programming task have some reference value to another. In this case, the results page of the programming task can be recommended as a revisited page.

The difficulty in recommending such revisitable web pages lies in figuring out the conditions under which pages are effective for the user's ongoing web browsing behavior.

2. Related Work

In the first part of this section, we summarize the findings from several studies on existing web page recommendations. In the second part, we analyzed some studies on web page revisitation. Finally, we discussed some web usage related research also including URLs studies.

2. 1. Web Page Recommendation

SASRec model [8] predicts the next items by seeking to identify which items are relevant from a user's action history based on self-attention. Top-N sequential recommendation [16] predicts the top-N ranked items that a user will likely interact with next by modeling each user as a sequence of items that interacted with in the past. Real-life recommendation systems often base only on short session-based data instead of long user histories, Bal ´ azs Hidas [6] is propose an RNNbased approach for session-based recommendations in which more accurate recommendations can be provided. Modeling the dynamic preferences of users is also important and challenging for recommendation systems. BERT4Rec [15] improves on previous recommendation methods that modeled user behavior in a one-way fashion from left to right and employ deep bidirectional self-attention to model user behavior sequences.

2. 2. Web Page Revisitation

Browsers support revisits with various tools, including bookmarks, history views and URL auto-completion. However, these tools only support revisits to a small number of frequently and recently visited pages. Several browser plugins and extensions have been proposed to better support the long tail of less frequently visited pages, using recommendation and prediction techniques [12]. A work in [18] constructs a predictive model to determine whether a website will be revisited by a particular user, the model can be used to filter web records to only present revisiting web pages. Study in [11] presents that the term page revisit had to be differentiated, and also identifies different types of revisitation that allow assessing the quality of current user support and developing concepts for new tools. A work in [7] leverages human natural recall processes, and proposed a personal web revisiting technique through contextual and semantic memory cues to facilitate recall. They discussed underlying techniques for context and content memories' acquisition, storage, decay, and utilization for page re-finding, also have added a relevant feedback mechanism to accommodate individual memory strength and revisiting habits.

The problem of the next-page prediction has been extensively studied in the literature. The method that has prevailed in this field, at least in terms of popularity, is Association Rules Mining. Association rules (AR) constitute a well-established method for effectively identifying related resources without taking into account their order of appearance (e.g., pages that are typically visited together, in the same session, but not necessarily in the same order) [3, 4]. Numerous works have investigated the performance of different variations of AR [2, 23, 37, 13, 31].

The dominant approach to predicting revistation, at least in terms of popularity, is association rule mining. Association rules (AR) are a well-established technique for efficiently identifying related resources regardless of the order in which they appear [2]. Several studies have examined the performance of different variants of association rules [10] [3].

2. 3. Web Usage

One of the earliest studies on Web usage behavior was in 1995 [17]. Tauscher and Greenberg found that 58% of an individual's pages are revisits and Web users often carried out recurrent tasks on the Web. Adar [1]investigated revisiting behavior by using a large user base collected via the Windows Live Toolbar. It was found that short-term browsing included basic browsing, visits to shopping or reference sites or information tracking pages. Medium-term visits included popular homepages, web mail, forums, educational pages, and browser homepages. Long-term visits included search engines and weekend activities, such as going to the movies.

URLs are also an important part of web usage and we discuss some URLs studies as well. Several studies detect malicious URLs using machine learning. Traditionally, Malicious URLs detection is done through the usage of blacklists, which cannot be exhaustive and cannot detect newly generated malicious URLs. URLNet [9] applies Convolutional Neural Networks to characters and words in URL strings, capturing several types of semantic information in URLs. R. Vinayakumaras [13] evaluate various deep learning architectures specifically Recurrent Neural Network (RNN), Identity-Recurrent Neural Network (I-RNN), Long Short-Term Memory (LSTM), Convolution Neural Network (CNN), and Convolutional Neural Network-Long Short-Term memory (CNN-LSTM) architectures by modeling the real known benign and malicious URLs in character-level language, finally, find out LSTM and the hybrid network of CNN and LSTM performance the best. URLdeepDetect [14] proposes a hybrid deep-learning approach for time-of-click URL analysis and classification to detect malicious URLs, also, determine a given URL as either malicious or benign by analyzing semantic and lexical features of a URL by applying various techniques, including semantic vector models and URL encryption. The PC Log used in this study is the browser's web browsing history, with the main object being the URL. Meanwhile, deep learning is also applied in the analysis of the system log.

Just as the detection of malicious URLs is helpful for users to use web services safely, the detection of anomalies in system logs is a key step to resuming a secure and trustworthy system. DeepLog [5] is a deep neural network model that utilized Long Short Term Memory (LSTM), models system logs as natural language sequences and detects anomalies when the log content deviates from the normal straight-line direction by learning the normally executed log content.

3. Preparation for Revisiting web Pages

3.1. Problem Definition

This research addresses the problem of recommending revisiting web pages from the currently browsing web pages. The inputs in this study are the most recently viewed web pages. The outputs are worth revisiting web pages. The worth revisiting web pages are supposed to be a reference to the most recently viewed pages.

Input Currently Browsing web Pages

Output Revisiting web Pages

For example, as shown in figure 1, if input the sub-pages

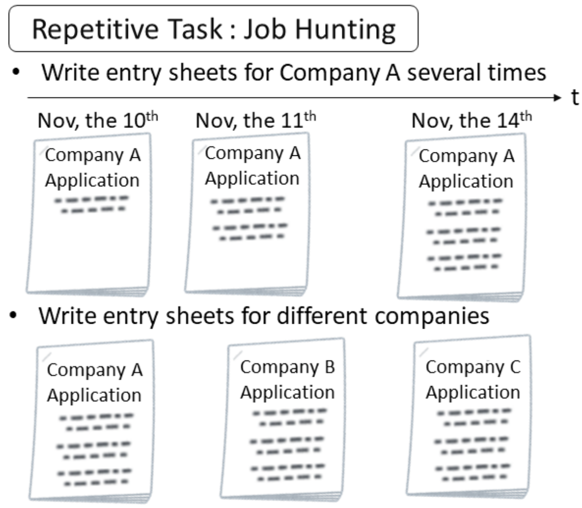

Figure 2 Repetitive task example

of the homepage of Company A, we aim for web pages with members of Company A's reviews and web pages with application form examples about Company A.

3.1.1. Currently browsing web pages

In this study, the pages with the newest records of the browsing history are defined as the currently browsing web pages.

3.1.2. Repetitive tasks

This study argues that people use computers to process repetitive tasks, that when Currently Browsing web Pages are similar or related to tasks that have occurred in the past, the Currently Browsing web Pages and a specific past session are defined as related sessions. The Revisiting web Pages should be in the browsing history, and also in the relevant session.

3.1.3. Revisiting web pages

To describe what revisiting pages are, it's necessary to describe the relationship between the signature web activities in the repetitive tasks and the revisiting page as shown in Figure 3.

Let us assume that a user planned to go to a 3-day-last music festival. The user's favourite singer G would give a performance at the music festival, but he didn't know which day it was. So the user found singer G by processing the schedule page. After a while, he was not sure which day his favourite artist would come, so he or she opened the festival website again. This time the user found singer G in the list of performers and then access the page with the order of the singer's appearances. Twice the user's web activities are shown in the figure 4.

The parts without marks are not the same, but twice of the web activities both start as parts green marked and end with parts yellow marked. Although the user processed singer G's schedule in different ways, the starting activities and ending

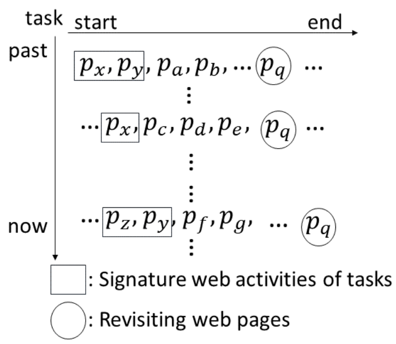

Figure 3 Revisiting web pages example

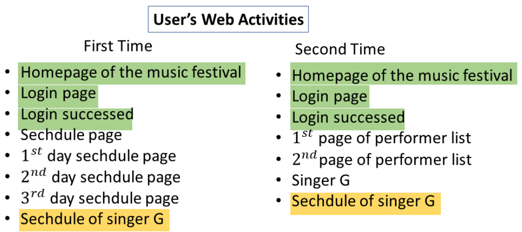

Figure 4 User's web activities

targets are the same.

Actions such as green, are called the signature web activities in the repetitive tasks. Target pages, like yellow, which is always reached after the signature web activities, is defined as revisiting pages. The method we propose is to skip the search process and go directly from the signature web activities to the revisiting pages.

3. 2. Revisiting Candidate Data

As described in 3.1.3, we collected pairs of data from the signature web activities and revisiting pages. In this section, we will discuss the metadata used and the processing methods.

3. 2. 1. Browsing history

The data used in this study is the web browsing history of the chrome browser. There is a number of data in the browse history database, we use only five of those data as follow.

- id
- url
- title
- visit time
- from id

The id is the default setting in chrome, each id appears only once and increases by 1 in chronological order. The URL and title are obviously the URL and title of the page being viewed.

The visit time is the time when the page was opened. The from id means that a page was opened by clicking on a link from another page. It should be 0 or less than the id of the browsing record itself. For example, if the SERPs after a search for a restaurant on google select one of the returned

id	url	title	visit_time	from_id
59165	https://uhgatet.u-hyogo.ac.jp/f5-w-...	UNIVERSAL PASSPORT RX	2022-07-24 17:50:56.260082	59163
59166	https://uhgatet.u-hyogo.ac.jp/f5-w-...	UNIVERSAL PASSPORT RX	2022-07-24 17:54:22.997730	59165
59167	https://uhgatet.u-hyogo.ac.jp/f5-w-...	UNIVERSAL PASSPORT RX	2022-07-24 17:54:23.271481	59166
59168	https://uhgatet.u-hyogo.ac.jp/f5-w-...	UNIVERSAL PASSPORT RX	2022-07-24 17:54:30.429293	59167
59169	https://uhgatet.u-hyogo.ac.jp/vdesk...	BIG-IP logout page	2022-07-25 00:28:51.989746	0
59170	https://uhgatet.u-hyogo.ac.jp/vdesk...	BIG-IP logout page	2022-07-25 00:28:52.031319	59168
59171	https://uhgatet.u-hyogo.ac.jp/my.po...	uhgatet.u-hyogo.ac.jp	2022-07-25 00:31:16.428326	0

Figure 5 Chrome web Browsing History

id	url	title	session_id
59165	https://uhgatet.u-hyogo.ac.jp/f5-w-687474707	UNIVERSAL PASSPORT RX	2878
59166	https://uhgatet.u-hyogo.ac.jp/f5-w-687474707	UNIVERSAL PASSPORT RX	2879
59167	https://uhgatet.u-hyogo.ac.jp/f5-w-687474707	UNIVERSAL PASSPORT RX	2879
59168	https://uhgatet.u-hyogo.ac.jp/f5-w-687474707	UNIVERSAL PASSPORT RX	2879
59169	https://uhgatet.u-hyogo.ac.jp/vdesk/hangup.ph	BIG-IP logout page	2880
59170	https://uhgatet.u-hyogo.ac.jp/vdesk/hangup.ph	BIG-IP logout page	2880
59171	https://uhgatet.u-hyogo.ac.jp/my.policy	uhgatet.u-hyogo.ac.jp	2881

Figure 6 Browsing History with session id

search results P1, and clicks on it, then the from id of P1 will be the id of the google search restaurant page. In this example, the google search restaurant page opens directly and is not clicked from another page, so its from id is 0. After the pre-processing of the dataset, 108,190 web browsing histories were obtained for this study.

During the data collection period, chrome was set as the default browser to ensure that there was no diversion of browser usage, we do this to ensure that as many pages viewed on the computer as possible were collected over the entire period. The data is collected in seven months from April 2022 to January 2023.

3. 2. 2. Session splitting

The purpose of splitting a session is to split the huge browsing history into several collections, using the collection of web pages to represent a particular task that the user is concentrating on. In this study, a session is considered to have ended when no new pages are opened for more than two minutes. Pages that are used in the same session are considered to have the same purpose, for example, to process the same task.

We calculate the time interval between each of the two adjacent historical visits to the web page by using Visit Time, split the session by determining whether the time interval is greater than 2 minutes, and attach the session id information. Add one after the other in chronological order. Finally got 5083 sessions in total.

3. 2. 3. Collecting Revisiting Candidate Data

In this section, we will describe what revisiting candidates are and how to find them in each session.

As showing in Figure 7, These are sessions with a time interval, and it can be known from the browsing history on January 15 that the user was doing something related to job hunting when this session occurred. After visiting the homepage of company A and the subpages in the homepage, the user opened the job hunting related website and searched for

Table 1 Sample of Revisiting Candidate Data

Action	target
Long-term transaction - Google Search	Learning long-term dependencies in NARX recurrent neural networks
python dataset splitting - Google Search[SEP] Evaluation and data set partitioning	Training, Validation, and Testing with BERT | Sentence Classification with BERT and Transformer
Company A - Google Search[SEP] Information of Company A	Example of entry sheet for Company A

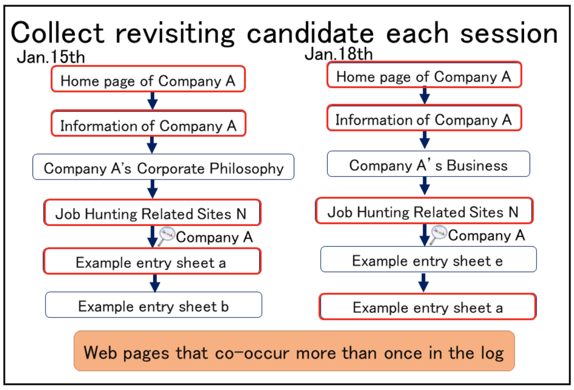

Figure 7 Collecting revisiting candidate

company A, trying to find some reference information, and finally reached the page of example entry sheet b.

It is important to note that there are many subpages in the company homepage and job search related websites, especially when searching for a company in job hunting related websites, just like search engines, the reference content returned may not be the same every time, and it takes time to find what you can refer to among hundreds of es example entry sheets.

The browsing history on January 18 shows that the user searched for company A on the job hunting related website after visiting company A's homepage, and again visiting page A. In the two visit records, the parts marked in red are the company homepage, the company information page, the job hunting related website and the content of company A on the job hunting related website. These four pairs of pages are called revisiting candidates, also the last one in the time series is Target, eg.Example entry sheet a, and the others are called Action, eg. Homepage of Company A, Information of Company A, Job Hunting Related Site N.

We collected 216 pairs of revisiting candidate using the approach described above, some examples of revisiting candidate data we collected in this study are shown in Table 1.

4. Revisiting Recommendation

As showing in Figure 8, we recommend revisiting pages

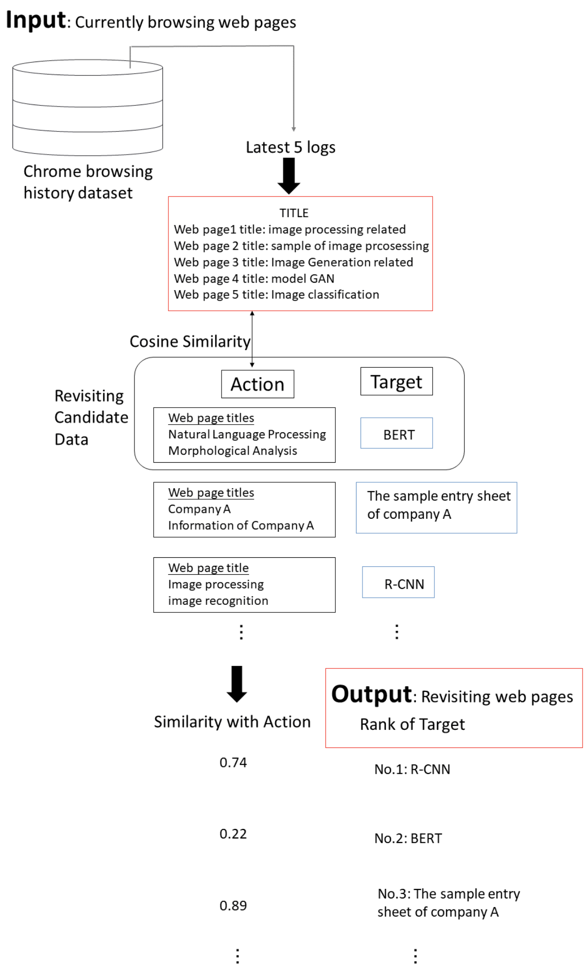

Figure 8 Method of revisiting recommendation

by calculating the similarity between the title of currently browsing pages and the signature web activity pages. In this section, we will discuss the process of similarity calculation and revisiting recommendation.

4. 1. Recommendation Interface

To implement revisiting web pages based on the web pages being viewed, the recommendations need to react timely. A simple recommendation interface is constructed to prompt the user for recommendations. As shown in Figure 9, the interface is linked to a database of chrome browsing history, and when the "search" button is pressed, the latest browsing history will be obtained for analysis. The recommended results will be output on the right side of the interface and pressing the "Open" button to open all.

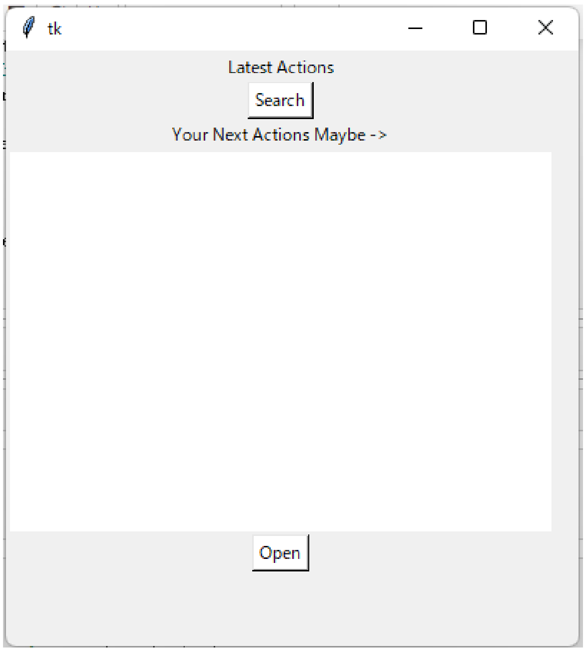

Figure 9 recommendation interface

4. 2. Vectorization of Title

Bidirectional Encoder Representations from Transformers (BERT) is a family of masked-language models published in 2018 by researchers at Google [4]. We used BERT word embedding to vectorize the title text, trying to get relevant sessions by cosine similarity calculations and recommending suitable pages. The steps are as follows:

- Keep only the title text of the page
- Recurring title elements are retained only once
- Combine the title elements into one line like a sentence with spaces
- Tokenize texts in each session
- – Get the session format for vectorization
- Vectorise each session
- Use cosine similarity to find the similarity between individual sessions

In this section, we construct a simple search system. The query is for a particular session and returns the top5 sessions with a high degree of similarity to it.

The returned session will basically have multiple occurrences of the words in the query or related words, but some do not have the same words or even words with similar meanings.

The returned sessions basically have multiple occurrences of the words in the query or related words, but some do not have the same words or even words with similar meanings. Also among the returned results containing related and similar words, it shows that the top results do not actually contain revisited pages that can be used. The results of calculating the similarity of texts are not satisfactory, so we tried to collect more data and try to get more accurate results by having the model learn the data from the relevant sessions of the positive solution.

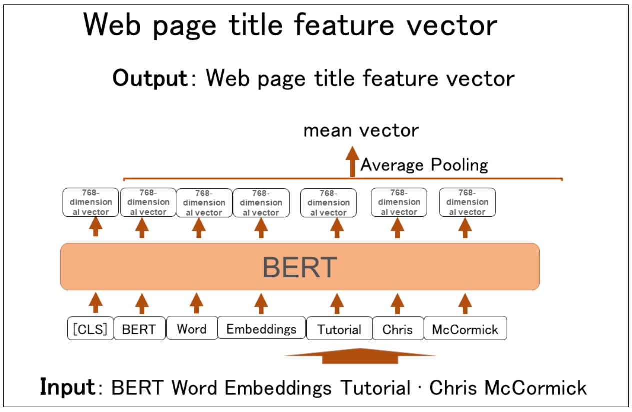

Figure 10 recommendation interface

4. 3. Revisiting Page Recommendation

In this section, We will use similarity calculations to recommend revisited pages based on the page being viewed. The flowchart of recommending the revisiting pages is shown in Figure 8.

The data used in this study was extracted from the browsing history database of chrome on the computer. Whenever a user opens a new web page, the database is updated with a new log. Therefore, for the pages being viewed, we use the 5 most recent recommendations for revisiting this database in chrome browsing. The data is just extracted one by one as shown in the figure. First, we format the output for the five latest logs and use the title of them. For example, the currently browsing web pages in Figure 8 are finally linked together like a single sentence. We treated the title as an input sentence.

Then, we calculate the similarity between currently viewing web pages and each of the Actions of revisiting candidate data. After generating drew the feature vectors of the currently browsing web pages and actions of revisiting candidates being visited, calculate the cosine similarity between the two vectors. From this, we can get the similarity between the currently browsing web pages and each action of revisiting candidates, to get a descending ranking of the revisiting candidates based on this similarity score. In this study, we work on finding browsing behaviors with the same purpose. And the targets in revisiting candidates with high similarity ranking are seen as useful web pages to the currently brows-

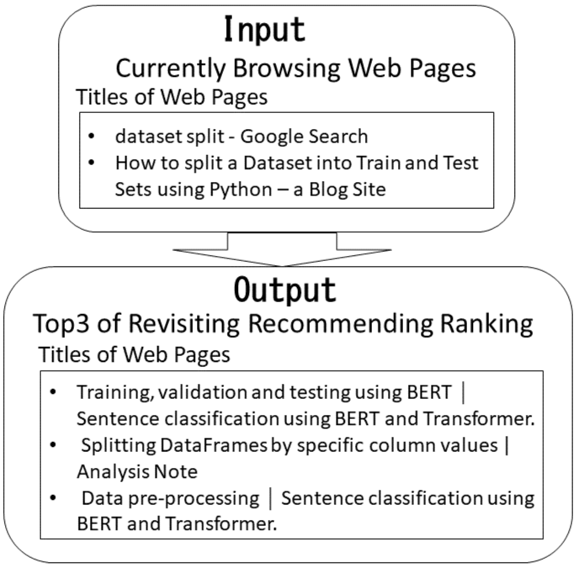

Figure 11 Revisiting Recommendation for Programming Task

ing web pages. Therefore, it can also be said that the targets of revisiting candidates with high similarity can be recommended as revisiting web pages because they have the same purpose. Based on the ranking of the revisiting candidates generated above, the target of revisiting candidates is output as revisiting web pages recommendation to the user.

5. Evaluation

In this section, we will examine several specific output examples of revisiting web pages recommendation.

Figure 11 shows a sample of revisiting web pages recommended by our systems, when searching for some coding problems in the programming task.

The programming task is for searching a coding problem, the approach of the segmentation dataset. All of the top3 recommended web pages are from past browsing history, and are relevant to dataset segmentation and pre-processing. Each of them is informative about the aim of the task. However, due to the limitation of training data, very few revisited pages could be recommended.

Figure 12 shows another sample of revisiting recommendations for job-hunting tasks.

The job-hunting task is for submitting Entry sheets to apply to companies, by referring to the intelligence on the company's official website and series of job-related websites. When the current browsing reaches the home page of Company A, the top3 revisiting web pages recommended by our systems are all very related to Company A on job-hunting websites. However, only pages that have been visited in the past can be recommended, so the range of job-related websites is very narrow.

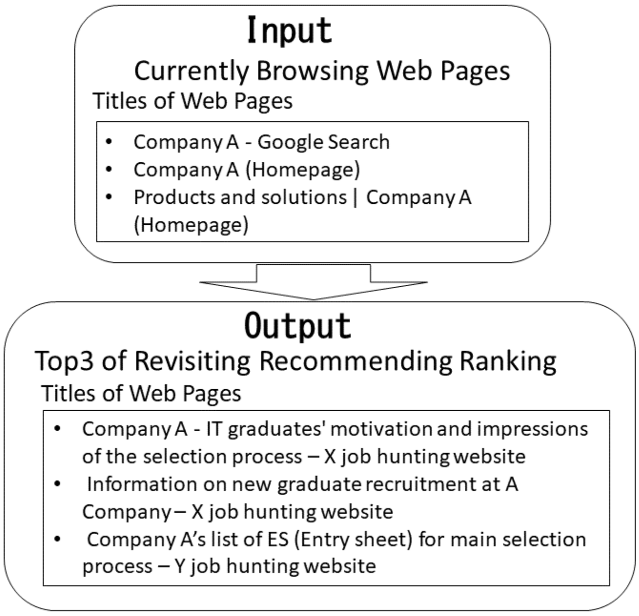

Figure 12 Revisiting Recommendation for Job Hunting Task

6. Conclusion

This paper proposes a system to automatically recommend revisiting web pages based on currently browsing pages. Revisiting web pages should be included in the pages that the user has visited in the past. As for the research process, we first defined signature web activities and revisiting pages. The recommendation for revisiting web pages consists of two steps. First, we collect revisiting candidate data based on the definition of signature web activities and revisiting pages. Then, by calculating the cosine similarity between currently browsing web pages and signature web activities, we obtain a ranking of recommending revisiting web pages. Finally, We evaluate our methods through some case studies. We found that this method currently suffers from limited revisited candidate web pages. The results are very informative in the face of revisiting behavior with a high degree of similarity. However, because the candidate revisiting data is limited, for most currently browsing web pages that do not have a high degree of similarity, the recommendation results are not informative for pages being viewed that are not very similar.

We consider making a more convincing evaluation in the future.

- Evaluation of different pre-treatment methods.

– Try other time intervals and observe the final results obtained for each time interval. The current threshold of 120s is artificially determined, which can actually be more scientifically and logically set to a value. For example, the bar chart of the time interval can be analyzed to have a numerical understanding of the end time of each network access behavior by observing the trough values roughly.

- Evaluate the output of revisiting pages.

– For this part of the evaluation, we are considering using the last month of the browsing history data set for the evaluation, and use the data from the very beginning to the beginning of the last month to give the model for learning. The data pre-processing of the validation set is still fine as it is now, but the data of the evaluation set needs to be more finely divided into sessions, and the selection of targets, i.e., the production of the positive solution data needs to be improved. In order to ensure the accuracy of the validation set data, it can be produced manually. Finally, the performance of the system on revisiting web page recommendations is evaluated by verifying whether the system's revisiting recommendations for the last month of web pages are consistent with the prepared positive solution data targets.

- Conducting user experiments.

– We are considering to asking multiple experimenters to use the system and experimenters, and also asked the experimenters to conduct a questionnaire on the feeling of using the system and the usefulness of the recommended results. In this, what is not clear yet is the overall system impression. For example, should our system be an interface resident on the browser, or can we find the right time to pop up recommendations by automatically determining which web page the user is currently browsing. The focus of these two types and the overall sense of the use of the system will be very different, in future work, we are committed to first determining the shape of the system, a more specific design of the overall definition and approach.

Future work includes but is not limited to improving accuracy by improving the existing recommendation system, or by more detailed pre-processing of web page browsing logs (e.g., segmenting sessions). We are also considering adding other computer logs as web activities to make more possibilities for revisiting recommendations.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Numbers JP21H03775, JP21H03774, and JP21H03554.

References

[1] Eytan Adar, Jaime Teevan, and Susan T Dumais. Large scale analysis of web revisitation patterns. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 1197–1206, 2008.

[2] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pages 207–216, 1993.

[3] Robin Burke, Michael P O'Mahony, and Neil J Hurley. Robust collaborative recommendation. *Recommender systems handbook*, pages 961–995, 2015.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[5] Min Du, Feifei Li, Guineng Zheng, and Vivek Srikumar. Deeplog: Anomaly detection and diagnosis from system logs through deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, page 1285–1298, 2017.

[6] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.

[7] Li Jin, Ling Feng, Gangli Liu, and Chaokun Wang. Personal web revisitation by context and content keywords with relevance feedback. *IEEE Transactions on Knowledge and Data Engineering*, 29:1508–1521, 2017.

[8] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining*, pages 197–206, 2018.

[9] Hung Le, Quang Pham, Doyen Sahoo, and Steven CH Hoi. Urlnet: Learning a url representation with deep learning for malicious url detection. *arXiv preprint arXiv:1802.03162*, 2018.

[10] Bamshad Mobasher, Robert Cooley, and Jaideep Srivastava. Automatic personalization based on web usage mining. *Communications of the ACM*, 43:142–151, 2000.

[11] Hartmut Obendorf, Harald Weinreich, Eelco Herder, and Matthias Mayer. Web page revisitation revisited: implications of a long-term click-stream study of browser usage. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 597–606, 2007.

[12] George Papadakis, Ricardo Kawase, Eelco Herder, and Wolfgang Nejdl. Methods for web revisitation prediction: survey and experimentation. *User Modeling and User-Adapted Interaction*, 25:331–369, 2015.

[13] Soman Ravi, Vinayakumar Kp and Prabaharan Poornachandran. Evaluating deep learning approaches to characterize and classify malicious url's. *Journal of Intelligent & Fuzzy Systems*, 34:1333–1343, 2018.

[14] Abdul Rehman Javed Mirza Beg Thar Baker Sara Afzal, Muhammad Asim. Urldeepdetect: A deep learning approach for detecting malicious urls using semantic vector models. *Journal of Network and Systems Management*, 2021.

[15] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450, 2019.

[16] Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, page 565–573, 2018.

[17] Linda Tauscher and Saul Greenberg. How people revisit web pages: Empirical findings and implications for the design of history systems. *International Journal of Human-Computer Studies*, 47:97–137, 1997.

[18] Hapnes Toba, Christopher Starry Jomei, Lotanto Setiawan, Oscar Karnalim, and Hui Li. Predicting users' revisitation behaviour based on web access contextual clusters. In *2020 8th International Conference on Information and Communication Technology*, pages 1–6, 2020.