Article

Extracting Searching as Learning Tasks Based on IBRT Approach

Pengfei Li 1, Bin Zhang 2 and Yin Zhang 2,∗

1 School of Computer Science and Engineering, Northeastern University, Shenyang 110167, China; leepengfei.neu@gmail.com
2 Software College, Northeastern University, Shenyang 110167, China; zhangbin@mail.neu.edu.cn
* Correspondence: zhangyin@mail.neu.edu.cn

Abstract: With the rapid development of the World Wide Web and information retrieval technology, learning supported by searching engines (such as making travel plans) has boomed over the past years. With the help of search engines, learners can easily retrieve and find large amounts of information on the web. Recent research in the searching as learning (SAL) area has associated web searching with learning. In SAL processes, web learners recursively plan tasks, formulate search queries, obtain information from web pages, and change knowledge structures, to gradually complete their learning goals. To improve the experiences of web learners, it is important to accurately present and extract tasks. Using learning styles and similarity metrics, we first proposed an IBRT model to implement structured representations of the SAL process for each learner. SAL tasks were then extracted from the structures of IBRT. In this study, a series of experiments were carried out against assignment datasets from the Northeastern University (China) UWP Programming Course. Comparison results show that the proposed method can significantly improve the performance of SAL task extraction.

Keywords: searching as learning tasks; improved Bayesian rose tree; hierarchical model

1. Introduction

Search engines are the most used tools for accessing online information and are increasingly being used for learning purposes, such as when making travel plans and finding properties to purchase. From the aspect of searching as learning (SAL), searching is conceptualized as a series of activities for learning, such as planning tasks, issuing queries, analyzing web sources, piecing information together, and evaluating and using information [1,2]. Unlike traditional studies on searching as a learning tool, which only focus on accurate search results, SAL focuses on learning activities and outcomes during the search process. Studies have determined that tasks are interleaved during SAL processes. Web learners must recursively plan learning tasks, formulate search queries, scan search results, and adapt their knowledge structure to complete learning tasks [3,4].

Therefore, accurate task extraction is important for information retrieval systems to support and foster a large variety of more sophisticated search strategies. Studies have shown that task extraction can help systems discover related and interesting tasks [5,6], improve learners’ engagement [7], and improve ranking performance [8]. Moreover, the accurate extraction of tasks also helps to accurately locate users’ learning statures in their task space, as well as further recommend and help users better finish their learning tasks [9]. In view of this importance, SAL task extraction has received considerable attention in recent years [10–12].

While some of these studies focused on task extraction, SAL task extraction is still a very challenging problem because systems struggle to quantify the concurrent learning during searching, each task can be defined at different levels of granularity, and learners have different learning styles [13]. Studies of constructivist epistemology have determined that, during the learning process, each learner must (i) identify the knowledge gap, (ii) plan search tasks, (iii) generate queries on the basis of their knowledge structure and learning...
style, and (iv) analyze and synthesize search results to improve the knowledge structure, recursively performing (i) until they complete the learning goal. From this perspective, learners with different learning styles demonstrate different planning preferences when developing search tasks, and these planning preferences lead to different issued queries. From the perspective of information retrieval system studies, the research and development of task extraction methods are always based on the query–query or query–results relationship. These methods may work well for search tasks such as look-up, but not necessarily for SAL tasks, as these require the searcher to reflect, learn, and analyze the results [14,15].

We are motivated by the need for an accurate representation and extraction of SAL tasks and propose an efficient improved Bayesian rose tree (IBRT) model. Most existing hierarchical task extraction methods cannot capture learning styles and SAL factors. In contrast, our proposed method can accurately represent tasks by capturing SAL factors and improve performance by employing learning styles. To test our method, we collected and analyzed students’ development processes from the UWP programming course at Northeastern University, where 46 students learned by searching the web to complete UWP programming assignments.

Collecting students’ periodic learning outcomes is difficult in real time, and manual observation of searching processes may introduce privacy concerns. Therefore, we designed the experimental environment based on two methods for collecting student trace data: (i) we developed an IDE plugin that can automatically collect snapshots; (ii) we developed an automated approach that can be applied to search behavior captured by a Chrome browser plugin. These methods help provide insight for a better understanding of how novice programmers go about developing search tasks. We found two learning styles in SAL through our analysis and used these insights to develop the IBRT model for SAL task representation and extraction. Specifically, we make the following contributions presented in this paper:

- We successfully collect an SAL dataset of autonomous web learner trace data.
- To our best knowledge, we are the first study to employ specific learning styles in SAL task studies.
- We develop a novel efficient SAL task representation and extraction based on an extended Bayesian rose tree model.

2. Related Works
2.1. Searching as Learning

Research in the field of SAL either conceptualizes learning as a searching process or devotes itself to exploring the connection between search and learning. Most of the studies that deal with learning in the search process do not conceptualize the phenomenon as learning; however, they observe changes in the search process, which can be interpreted as learning [2]. The concept of SAL was first pointed out in the Dagstuhl seminar (2013) [14] and formulated a strong research agenda for follow-up studies. Reynolds et al. [16] figured out that e-learning and information retrieval may provide useful boundaries and definitions for SAL.

(1) From the perspective of conceptualizing learning as a search process. Kuhlthau [17] first conceptualized learning as a search process and proposed the Information Search Process (ISP) model. This model emphasizes the stages of human cognition and the learning process. Gary and Marchionini [15] pointed out that searches for learning purposes involve multiple iterations, and learners must analyze and interpret search results to achieve the learning goal. Zarro [18] presented dual-process theory from the social psychology domain conceptualizing learning as searching behaviors and cognitive processes. On the basis of conceptualizing learning as a search process, Odijk and White [19] figured out that learners may appear to be struggling during the learning process. Rieh et al. [1] assessed learning from online searching behavior and develop a search system that supports SAL. Ghosh et al. [16] studied the relationship between search and learning by taking learning as the result of the information search
process. Proao-Rí et al. [20] conceptualized searching as a learning path and focus on helping learners to plan their SAL path. Taibi et al. [21] analyzed search processes in web learning and proved the positive impact of SAL systems in stimulating students’ creativity and critical thinking.

(2) From the aspect of exploring the relations between search and learning. Vakkari et al. [2] summarized existing studies and presented three types of relationships between searching and learning. Moraveji et al. [22] and Sun et al. [23] analyzed the relationship between search skills and learning outcomes and pointed out that search skills can help learners complete learning tasks in a shorter time. Liu and Belkin [24] pointed out that learners’ familiarity and experience with topics will affect search behaviors and further positively affect learning outcomes. Bron et al. [25] and Vakkari et al. [26] also focused on how variables in the search processes affect search results and learning outputs. Gimenez et al. pointed out that searching is the learning mode of learners in the learning process, which will affect the search process [27]. Marchionini [28] focused on the relationships among searching, sense-making, and learning. Yigit et al. [29] explored result diversification as a useful technique to support learning-oriented search and developed a new search engine for SAL. Liu et al. [30] determined that learning strategies will directly affect their learning outcomes, and they further presented two kinds of learning strategies; they found that learners with a task-adaptive strategy show better learning outcomes, e.g., knowledge points, facets, and scope.

2.2. Task Extraction

Task extraction is an important research topic in both information retrieval and e-learning [31–33]. Wang et al. [34] pointed out that, in order to complete complex tasks (such as learning), users usually need to submit a series of queries, and multiple sub-learning tasks are involved in the process. If the search engine can predict multiple sub-targets of the learning task, the user can complete the task efficiently.

Traditional SAL task extraction methods usually start from two aspects: (1) SAL behavior duration; (2) exploring relationships (e.g., the relationship between query and query, the relationship between search results and search results, or the relationship between learning content and learning content). While some of these studies did not clearly point out the aspects for learning tasks, they can be used to identify SAL tasks. The methods which focus on SAL behavior duration extract tasks on the basis of the time between queries. If the time interval between two consecutive queries exceeds a certain threshold, it will be regarded as two different sessions or tasks; usually a 30 min timeout period is typically used to subdivide the session [35]. SAL task extraction methods based on exploring relationships mainly focus on classification and clustering methods to extract learning tasks from query sequences. The common method of SAL task extraction is to calculate the distance between queries. Kotov et al. [36] analyzed cross-session search behavior and proposed a method for identifying all related queries from previous sessions issued by a user. Li et al. [37] proposed a task extraction method to identify search tasks. The studies that explored the relationship between learning content mainly abstracted queries as knowledge or concepts. Mittal et al. [38] abstracted queries into concepts using relational graph theory and explored the connections between concepts.

In addition, some advanced studies used hierarchical models to identify the relationships between tasks. Most of the existing hierarchical clustering techniques produced a binary tree structure, where each node was decomposed into two sub-nodes [13]. Jones et al. proposed and verified a method of automatically dividing the user query flow into hierarchical units, which automatically divides the user query flow into hierarchical tasks [39]. Blundell [40] first proposed the Bayesian hierarchical structure model to realize the hierarchical structure division of search tasks. Recently, Mehrotra et al. [13] proposed an efficient Bayesian nonparametric model to discover the hierarchical structure, and they clarified that the model can recognize the task structure composed of any number of subtasks.
3. Data and Methods

3.1. The Definition of Tasks in SAL

Jones et al. [15] first figured out the importance of task extraction and proposed that a task is the atomic information requirement that results in one or more queries. On the basis of Jones’s work, Ahmed et al. [18] proposed a more general definition of the complex web search task that captures the hierarchical structure of tasks and their associated subtasks. They also proposed that each search task can be divided into one or more subtasks which can also be considered independent tasks of the search and can eventually be decomposed into simpler tasks until each task represents the atomic information needs of the user. In addition, Ahmed et al. defined a complex web search as a multifaceted or multistep information requirement consisting of a set of related subtasks, each of which may recursively be considered a search task [18,41]. This definition is more general because it captures all kinds of tasks, whether complex or noncomplex.

Research by Rieh et al. [1] pointed out that it is very important to distinguish different types of learning structures in SAL. When the information is considered to be knowledge or belief from constructivist epistemology, searching will be conceptualized as a learning process. Users develop learning tasks and measure search results according to the conceptual changes of knowledge structure and beliefs, and they gradually change their knowledge structure to complete search-based learning tasks.

In addition, the authors of [2,26,42,43] pointed out that learners show different learning styles, which lead to different search preferences. Kolb et al. [44] classified learning styles into four categories: diverger, assimilator, converger, and accommodator; and this theory has become one of the most widely used learning models for students [45]. Moreover, he also proposed that investigating learners’ learning styles before a course can help teachers achieve better teaching effects [46]. On the basis of Kolb’s theory, Li et al. [47] proposed that learning styles affect the learning process (e.g., planning tasks) and further affect students’ learning outcomes. Vizeshfar et al. [48] pointed out that the understanding of learning styles helps to develop methods to improve students’ performance.

According to the above studies, we consider that the tasks in SAL cover the following characteristics:

- Hierarchical: Each process of SAL may cover one or more tasks, and each task may be divided into subtasks. In addition, subtasks can also be independent tasks. Therefore, an effective task extraction method should be able to accurately identify and represent this hierarchy.
- Learning style: In the process of SAL, learners divide learning goals depending on their learning style. Therefore, it is also important to develop a learning style-oriented SAL task extraction method.

On the basis of the definition of tasks in complex search (Ahmed et al. [5]), in this paper, we propose the definition of SAL tasks from the perspective of the tasks in SAL below.

Multifaceted or multistep learning activities are generated by learning needs; in the process of satisfying a learning goal, the learner may partition the learning goal into a set of tasks according to their own learning style, and each task may be recursive to be partitioned to a set of low-granularity tasks.

3.2. Data Collection and Analysis

3.2.1. Data Collection

For an understanding of how students develop SAL tasks, we collected and analyzed students’ searching and learning behaviors in the Northeastern University’s UWP course. First, we modified the Integrated Development Environment (IDE) for UWP to log students’ learning processes. Specifically, a snapshot of the student’s program was logged when the student compiled their project. Second, we developed a Firefox browser plug-in to log students’ searching activities. This plug-in was installed voluntarily by the student and could log complete searching interactions between students and search engines. To protect
the privacy of students, we removed information that may involve student privacy, and we also anonymized student information. For this investigation, we analyzed data from two types of assignments, described below.

(1) Assignments with clear targets. In the first week of the UWP course, students were asked to develop a multimedia player with file opening and playing functions. The full solution needed to work on the Win10 system and be able to play video and audio files. During this process, students struggled between searching and learning to finally complete the assignment. Similarly, we also assigned four other UWP programming assignments in the subsequent 4 weeks.

(2) Assignments with open targets. In the sixth week of the UWP course, each student was asked to develop a UWP application with open targets. In order to understand the actual learning task, each student was asked to submit their study reports in text form.

3.2.1. Data Collection

To extract tasks in SAL, we must be able to identify the learning styles. However, asking a learner about their learning style is not feasible in most cases. We believe that learning styles can be deduced by examining the learning activities. Therefore, we manually analyzed the learning activities of the students on UWP course. During our analysis, we observed two distinct learning styles. Figure 1a, b are two representative examples of SAL learning style from the dataset we collected.

3.2.2. SAL Learning Style Analysis

To extract tasks in SAL, we must be able to identify the learning styles. However, asking a learner about their learning style is not feasible in most cases. We believe that learning styles can be deduced by examining the learning activities. Therefore, we manually analyzed the learning activities of the students on UWP course. During our analysis, we observed two distinct learning styles. Figure 1a, b are two representative examples of SAL learning style from the dataset we collected.

Figure 1. Examples of SAL learning styles ((a) an example of the first learning style, (b) an example of the second learning style).

Figure 1a presents an example in which the student is learning to complete the first-week assignment. The student first issued a query “file open picker”. Combined with the click-on events and the context in the UWP website, we find that the query can be mapped to the UWP class “FileOpenPicker”. The second query “FileSavePicker” and the third query “FolderPicker” are the names of two classes in the UWP API. Through analysis, we found that “FileOpenPicker”, “FileSavePicker”, and “FolderPicker” are all classes under the namespace “Windows.Storage.Pickers”. Therefore, it is evident that the student was learning following the knowledge network on Microsoft’s official UWP API reference. In addition, we believe that these searching activities belong to a common task (learning how to open a file for programming the media player).
Figure 1b shows another learning style. During the SAL process, the student issued five queries following the time sequence. Combined with context on the Microsoft UWP website, we find that these learning events fall into three main learning tasks (the first and the second query belong to the first learning task, the second query and the third query belong to the second learning task, and the fourth query belongs to the third learning task). The analysis of the programming snapshots validated the deduction that the student completed three functions for the first-week assignment during this part of the learning process. Obviously, the student’s learning path did not follow the knowledge network of the UWP API reference.

The aforementioned examples represent the types of learning styles we observed during the data analysis. We summarized the two learning styles as follows:

1. Learning according to the knowledge network (LKN). These learners prefer to learn by following authoritative knowledge points in the knowledge network and creating queries on the basis of these knowledge points. In our experiments, these students commonly searched and learned following Microsoft’s official UWP API reference.

2. Learning according to information needs (LIN). Learners who learn according to information needs prefer to learn according to the tasks partitioned by their own subjective. Compared with LKN, LIN learners exhibited more interleaved tasks. In the UWP course, these students usually explored the knowledge points needed to complete the learning task and learned to program sequences.

3.3. Modeling Progress

The first step in our SAL task extraction modeling process was to learn a high-level representation of SAL tasks for each learner. To this end, we propose an improved Bayesian rose tree (IBRT) model for representation hierarchies of SAL tasks. While we cannot directly observe the task states when learning (they are latent variables), we can observe the search queries, search results, and the source code. These features serve as predictor variables for these latent variables. Search queries can reflect the learning objects in tasks, as well as possible learning outcomes, and the source code snapshot can reflect the task states. We propose a task extraction method to accurately extract tasks from the tree structure.

3.3.1. Improved Bayesian Rose Tree Model

The IBRT model is based on the BRT model used to construct multibranch hierarchies. Blundel et al. [40] first proposed an efficient Bayesian nonparametric model for discovering hierarchical community structure in social networks. BRT is based on a greedy probabilistic agglomeration approach to construct multibranch hierarchies. Mehrotray et al. [13] first proposed the BRT (Bayesian rose tree) model for extracting hierarchies of tasks and subtasks. Although the traditional BRT algorithm can achieve a hierarchical representation of search tasks, the BRT model cannot be used for SAL task representation as it cannot take learning outcomes and learning styles into account.

To solve these problems, we propose the IBRT model for the representation hierarchies of SAL tasks. IBRT can sense both searching features and learning features, further providing accurate SAL task representation on the basis of learning styles. The authors of [2,44] pointed out that the query is the starting point of a SAL process, and it can reflect learning objectives and learning structure. Furthermore, the authors of [2] proposed that queries can reflect the learning tasks of the learner. Therefore, we use query sets to represent the SAL task–subtask structures.

Specifically, we first propose Equation (1) in the IBRT model to build the representation of SAL tasks.

\[
p(D_m|T_m) = \pi_{LKN}g(D_m|T_m) + (1 - \pi_{LKN}) \left( \pi_{T_m}f(D_m) + (1 - \pi_{T_m}) \prod_{T_i \in \mathcal{d}(T_m)} p(D_i|T_i) \right),
\]
where $D_m = \{Q_1, Q_2, \ldots, Q_m\}$ is the set of issued queries. The queries in $D_m$ are sorted by the time sequence, where $Q_i$ is the $i$-th issued query. $p(D_m | T_m)$ is the probability of data $D_m$ given a partitioning by the tree $T_m$. $g(D_m | T_m)$ is the marginal probability for $D_m$ which is partitioned by $T_m$ when the user’s learning style is LKN. $\pi_{LKN}$ is the prior probability of the learners whose learning style is LKN, whereas $(1 - \pi_{LKN})$ is the prior probability of the learners whose learning style is LIN. $\pi_{Tar}$ is the mixture probability, which represents the prior probability that all data points in $T_m$ belong to a single cluster. $f(D_m)$ is the marginal probability of the data $D_m$. $p(D_i | T_i)$ is a hyperparameter representing the probability that the nodes in query set $D_i$ are partitioned given the tree $T_i$.

At the beginning of the modeling process, each query in $D_m$ is considered an independent tree. Specifically, $T_i = \{x_i\}$, where $x_i$ is the $i$-th query in $D_m$. A query may independently be a task or part of a task. We recursively define the representation of SAL task hierarchy, where a tree $T$ may contain queries $T = \{x_1, x_2, \ldots, x_n\}$ or may be split into subtrees $T = \{T_1, T_2, \ldots, T_n\}$, where each $T_i$ ($1 < i < n$) is a subtree in $T$. This allows us to consider $T$ as a hierarchical tree structure for the task–subtask relationship, where tasks and subtask are represented by query sets.

During the modeling process, for each step, the IBRT method chooses two trees $T_i$ and $T_j$ from $T$ and merges them into a new tree $T_m$. In this paper, we adopt the same tree merging strategy along with BRT, as shown in Figure 2. Inspired by the study proposed by Mehrotra and Yilmaz [13], at each step, IBRT greedily finds and merges two trees when the following formula is maximum:

$$\frac{p(D_m | T_m)}{p(D_i | T_i)p(D_j | T_j)},$$

(2)

where $p(D_m | T_m)$ is the possibility of the partition of query set $D_m$ given the tree $T_m$. All queries in $D_m$ are leaves of tree $T_m$, and $p(D_m | T_m)$ is calculated recursively through its child nodes.

![IBRT subtree merging strategy](image_url)

**Figure 2.** IBRT subtree merging strategy.

(1) Calculating the marginal probability for LKN

For learners whose learning style is LKN, $g(D_m | T_m)$ is calculated according to the knowledge network. As discussed above, in our experiment, students usually searched and learned using Microsoft UWP official documents. Thus, we propose calculating $g(D_m | T_m)$
according to the hierarchical association between classes in the application programming interface provided by Microsoft’s official website for UWP app development.

As discussed in Section 3.2.2, students always learned following the Microsoft API reference for UWP. Thus, we built a tree structure for the knowledge network on the basis of the API reference. Specifically, on the API reference, each class belongs to a unique namespace, and a namespace contains multiple classes. Accordingly, we built each namespace as a subtree of the tree structure, while the classes under the namespace constituted the leaves of the subtree.

All namespaces belonged to the set of Windows UWP namespaces. Therefore, these subtrees were used to construct a three-layer tree structure for the UWP knowledge network. In the tree structure, all UWP classes were leaves. Figure 3 shows the tree structure of the UWP API reference, where “Windows.UI.Xaml.Media” is a namespace that contains classes such as “AcrylicBrush”, and “ArcSegment”. To determine the consistency of the representation of the tree structure, we removed the nodes other than leaves in the tree for the knowledge network.

![Figure 3. The tree structure of the UWP API reference.](image)

Second, we mapped user queries into the knowledge network to build tree structures for each student. Specially, we mapped user queries to corresponding tree nodes in the tree of the UWP knowledge network by semantic similarity. To this end, we crawled websites about UWP programming to supplement our corpus and trained the word2vec model.

We calculated the semantic similarity from two aspects: (i) the semantic distance between queries and classes; (ii) the belonging of newly added classes (one of the learning outcomes) to the knowledge network tree structure. We propose Equation (3) for calculating the degree of association strength between user queries and classes such as “AcrylicBrush”, and “ArcSegment”. To determine the consistency of the representation of the tree structure, we removed the nodes other than leaves in the tree for the knowledge network.

\[
\text{Semsim}(q_i, \text{class}_j) = \cos(v(q_i), v(\text{class}_j)) + \beta \text{belong}(\text{class}_{q_i}, \text{class}_j),
\]

where \(v(q_i)\) is the semantic vector for \(q_i\), and \(v(\text{class}_j)\) is the semantic vector for \(\text{class}_j\). \(\text{belong}(\cdot)\) is an indicator function. When \(\text{class}_{q_i}\) and \(\text{class}_j\) belong to the same namespace, \(\text{belong}(\text{class}_{q_i}, \text{class}_j) = 1\); otherwise, \(\text{belong}(\text{class}_{q_i}, \text{class}_j) = 0\).

Lastly, \(g(D_m|T_m)\) is calculated by the tree edit distance between a student’s knowledge network tree and \(T_m\) according to the method proposed by Dinler et al. [49].
For the students who study according to their learning needs, this paper proposes the following method to calculate the marginal probability $f(D_m)$:

$$f(D_m) = \prod_{k=1}^{n} p \left( \sum_{i \in 1..|Q|} \sum_{j \in 1..|Q|} r^k_{q_i|q_j} |a_k \beta_k| \right), \quad (4)$$

where $r$ represents the degree of association strength between two nodes in the tree. We calculated $r$ on the basis of the SAL features which we extracted from students’ web search data and learning outcomes. Table 1 shows the influencing factors that we selected to calculate the degree of association strength between user-issued queries in the SAL process. We employed the method proposed by Mehrotra et al. [6] to calculate $\pi_{T_m}$ as follows:

$$\pi_{T_m} = 1 - (1 - \gamma)^{\text{child}(T_m)} - 1, \quad (5)$$

where $\text{child}(T_m)$ presents the number of children in the tree $T_m$, and $\gamma$ is a hyperparameter.

**Table 1. The influencing factors.**

<table>
<thead>
<tr>
<th>Description</th>
<th>Factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cosine similarity between term sets of two queries</td>
<td>$\text{cosin} (\text{TermSet}<em>{q_i}, \text{TermSet}</em>{q_j})$</td>
</tr>
<tr>
<td>Edit distance between term sets of two queries</td>
<td>$\text{edit} (\text{TermSet}<em>{q_i}, \text{TermSet}</em>{q_j})$</td>
</tr>
<tr>
<td>Jaccard distance between term sets of two queries</td>
<td>$\text{jacc} (\text{TermSet}<em>{q_i}, \text{TermSet}</em>{q_j})$</td>
</tr>
<tr>
<td>Percentage of identical UWP terms between queries</td>
<td>$\text{pct_uwp_terms} (\text{TermSet}<em>{q_i}, \text{TermSet}</em>{q_j})$</td>
</tr>
<tr>
<td>Cosine semantics distance between queries</td>
<td>$\text{sem_cosin} (\text{TermSet}<em>{q_i}, \text{TermSet}</em>{q_j})$</td>
</tr>
<tr>
<td>Search results</td>
<td></td>
</tr>
<tr>
<td>Average cosine similarity between term sets of clicked on URLs after two queries</td>
<td>$\text{avg_url_cosin} (\text{URLTermSet}<em>{q_i}, \text{URLTermSet}</em>{q_j})$</td>
</tr>
<tr>
<td>Average edit distance between term sets of clicked on URLs after two queries</td>
<td>$\text{avg_url_edit} (\text{URLTermSet}<em>{q_i}, \text{URLTermSet}</em>{q_j})$</td>
</tr>
<tr>
<td>Average cosine semantics similarity between clicked on URLs after two queries</td>
<td>$\text{url_avg_context_cosin} (\text{URLTermSet}<em>{q_i}, \text{URLTermSet}</em>{q_j})$</td>
</tr>
<tr>
<td>Cosine distance of UWP terms contained in clicked links after two queries</td>
<td>$\text{url_avg_terms_cosin} (\text{URLTermSet}<em>{q_i}, \text{URLTermSet}</em>{q_j})$</td>
</tr>
<tr>
<td>Cosine similarity between UWP class sets of two search results</td>
<td>$\text{avg_result_class_cosin} (\text{SearchResult}<em>{q_i}, \text{SearchResult}</em>{q_j})$</td>
</tr>
<tr>
<td>Learning outcomes</td>
<td></td>
</tr>
<tr>
<td>Cosine similarity between class sets of snapshots after two queries</td>
<td>$\text{snapshot_cosin} (\text{Snapshot}<em>{q_i}, \text{Snapshot}</em>{q_j})$</td>
</tr>
<tr>
<td>Edit distance between class sets of snapshots after two queries</td>
<td>$\text{snapshot_edit} (\text{Snapshot}<em>{q_i}, \text{Snapshot}</em>{q_j})$</td>
</tr>
</tbody>
</table>

IBRT adopted the influencing factors from three main aspects:

- **Queries:** (i) Queries with identical or similar terms tend to have the same learning tasks. We, thus, used this factor to capture task relationships between a pair of search queries. To this end, we adopted cosine similarity, edit distance, and Jaccard distance to calculate these relationships. (ii) Queries with the same UPW-related terms tend to belong to the same tasks. In this paper, we adopted ‘percentage of identical UWP terms between queries’ to capture this feature. (iii) Pairs of search queries that are closer in the semantic space tend to belong to the same task. We, thus, used the ‘cosine distance between query semantics’ to capture this factor.

- **Search results.** Search queries that belong to the same tasks tend to have similar search results. Therefore, search results are also an important factor for us to measure the relationships between a bunch of queries. Similarly, we used the ‘average cosine similarity between term sets of clicked on URLs after two queries’, ‘average edit
distance between term sets of clicked on URLs after two queries’ etc. to calculate these relationships.

- Learning outcomes. We used snapshots to analyze how learners develop tasks since snapshots are the bridge linking queries and learning outcomes. In this paper, we used the ‘cosine similarity between class sets of snapshots after two queries’ and the ‘edit distance between class sets of snapshots after two queries’ to calculate these relationships.

3.3.2. Extraction of the Tasks in SAL Based on IBRT

The final step in our algorithm was to find tasks through the IBRT structure. We employed the dataset collected from the UWP course. To find these tasks, we employed the task coherence score in the BRT. To calculate the task coherence score, we ran the method developed by Mehrotra [6]. Using this relationship measure, we identified whether a subtree was a task or should be divided into multiple tasks. For each student, we constructed an IBRT to represent their learning status. We then computed the task coherence score for each leaf in each level.

4. Experiment Evaluation

In this section, we describe the experimental setup and evaluate the proposed SAL task extraction method. We evaluate the proposed SAL task extraction method from two aspects. First, we compare the performance of IBRT with existing state-of-the-art task extraction methods on the manually labeled UWP dataset. Second, subjective metrics (opinions from the participants) are collected to evaluate the performance, such as prediction accuracy and task hierarchy structure of IBRT.

4.1. Experimental Setup

We performed several experiments to study the performance of our approach. We designed experiments using the data described in Section 3.2.1. We collected repositories with 402 programing assignments (335 of the programing assignments with clear targets, and 67 of them with open targets) from 67 Northeastern University students. For each student, we collected an average of 63 snapshots of their learning outcomes, and we logged an average of 102 queries and an average of 251 clicks on events (these data were time-stamped, and we anonymized students’ search logs and used the ZEALOUS algorithm to protect student privacy data [50]). Lastly, we collected a study report for each student.

In this paper, we also used the best-performing hyperparameters (in our method and baseline methods). Additionally, for $\pi_{TM}$, we employed the proportion of learning style LKN that we found during manual classification, $\pi_{TM} = 0.4$, which led to the best performance using our method, and this parameter was employed for all experiments in this paper. To evaluate our model, we then manually classified and annotated the UWP dataset, and all methods were trained on the dataset.

4.2. Comparison with State-of-the-Art Methods on UWP Dataset

To evaluate our model, we compared the results of the performance with seven existing state-of-the-art task extraction methods (BRT [13], BHC [51], LDA-Hawkes [37], Bestlink-SVM [52], Cluster-SVM [53], QC-HTC [54], and QC-WCC [54]). For a fair comparison, we experimented with these methods using the same data. Figure 4 compares the proposed model with the state-of-the-art task identification methods using F1 scoring.
In Figure 4, it is obvious that IBRT significantly outperformed other baseline algorithms. We first observed that BRT (to obtain a fair comparison, we employed all the factors in BRT that we employed in IBRT) performed worse than IBRT. The reason is that BRT cannot optimize according to students’ learning patterns. In contrast, in IBRT, we optimized two kinds of learning styles. QC-WCC and QC-HTC showed a difference in performance because they use different strategies in postprocessing.

4.3. Subjective Metrics

In this section, we applied subjective analysis to identify the performance of the proposed method.

We first discuss the subjective analysis setup. On the basis of the experiments we discussed in Section 4.1, we convened the participating students from the UWP course to conduct a subjective analysis of IBRT and other baseline methods. Each student was asked to evaluate their own search log. Students were asked to recall their own learning process and to evaluate the performance of the different algorithms. For a fair comparison, we anonymized the algorithm names, and we also shuffled the order of the extraction results when displaying them to students. Specifically, each student was asked to score the performance according to the following two aspects.

(1) The accuracy of SAL task extraction

For this, the participants were asked to score the accuracy of each method. To this end, each student was asked to answer the following question:

“Does the task extraction result match the real situation when you are learning?”

We list the scoring principles in Table 2.

Table 2. The scoring principles for question 1.

<table>
<thead>
<tr>
<th>Score</th>
<th>Scoring Principles</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Mismatches the real situation</td>
</tr>
<tr>
<td>1</td>
<td>Only a small amount matches the real situation</td>
</tr>
<tr>
<td>2</td>
<td>Mostly matches the real situation</td>
</tr>
<tr>
<td>3</td>
<td>Completely matches the real situation</td>
</tr>
</tbody>
</table>
(2) The accuracy of the hierarchy

For this, the participants were asked to evaluate the hierarchy of each method. To this end, each student was asked to answer the following question:

“Do you think the hierarchy of your learning process is valid?”

We list the scoring principles in Table 3.

Table 3. The scoring principles for question 2.

<table>
<thead>
<tr>
<th>Score</th>
<th>Scoring Principles</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Invalid</td>
</tr>
<tr>
<td>1</td>
<td>Valid in some parts; however, most of the extraction is invalid</td>
</tr>
<tr>
<td>2</td>
<td>Valid in some parts; however, some parts are invalid</td>
</tr>
<tr>
<td>3</td>
<td>Valid</td>
</tr>
</tbody>
</table>

Table 4 presents the evaluating results for the accuracy of SAL task extraction. A higher score indicates that the method had a better performance. The score of IBRT was the best with 2.7 (about 85% of students believed that the extraction results completely match their learning process). BRT performed second best with 2.5. This result also validates the performance of the BRT model in task extraction. In addition, the student evaluations were consistent with our experimental results in Section 4.2.

Table 4. Evaluating results for the accuracy of SAL task extraction.

<table>
<thead>
<tr>
<th></th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBRT</td>
<td>2.7</td>
</tr>
<tr>
<td>BRT</td>
<td>2.5</td>
</tr>
<tr>
<td>BHC</td>
<td>2.1</td>
</tr>
<tr>
<td>LDA-Hawkes</td>
<td>2.5</td>
</tr>
<tr>
<td>BestLink-SVM</td>
<td>2.4</td>
</tr>
<tr>
<td>Cluster-SVM</td>
<td>2.3</td>
</tr>
<tr>
<td>QC-HTC</td>
<td>1.9</td>
</tr>
<tr>
<td>QCWCC</td>
<td>2.2</td>
</tr>
</tbody>
</table>

Table 5 presents the evaluating results for the accuracy of the hierarchy. A higher score indicates that the method had a better hierarchy. The score of IBRT was the best with 2.3 (about 69% of students believed that the extraction results completely matched their learning hierarchy). BRT performed second best with 2.0. BHC presented the worst score, as it adopts a binary tree structure, which is usually not the case for real learning tasks.

Table 5. The evaluating results for the accuracy of the hierarchy.

<table>
<thead>
<tr>
<th></th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBRT</td>
<td>2.3</td>
</tr>
<tr>
<td>BRT</td>
<td>2.0</td>
</tr>
<tr>
<td>BHC</td>
<td>0.9</td>
</tr>
<tr>
<td>LDA-Hawkes</td>
<td>1.6</td>
</tr>
<tr>
<td>BestLink-SVM</td>
<td>1.4</td>
</tr>
<tr>
<td>Cluster-SVM</td>
<td>1.4</td>
</tr>
<tr>
<td>QC-HTC</td>
<td>1.5</td>
</tr>
<tr>
<td>QCWCC</td>
<td>1.5</td>
</tr>
</tbody>
</table>

5. Conclusions

Studies on SAL are promising for designing advanced searching and education systems. To improve the experience of web learners, it is important to accurately present and extract SAL tasks. On the basis of learning styles and similarity metrics, this study
proposed the IBRT model to implement structured SAL task representations for learners and, thereafter, extracted tasks on the basis of that structure. To test the proposed method, we collected and analyzed experimental data from the Northeastern University UWP course. Subsequently, a series of experiments were performed against the collected datasets. In comparison to the state-of-art task extraction methods, the experimental results indicated that our proposed methods could significantly improve the performance of SAL task extraction in terms of both objective and subjective metrics.

The success of the IBRT model implies that SAL task extraction studies combining both searching and learning activities are promising. In the IBRT model, searching activities can be conceptualized as a part of learning that further affect learning outcomes. The analysis and experimental results lead us to believe that the IBRT model can further explore the relationships between searching and learning activities.

Our results also demonstrate the advantage of employing learning styles in SAL task extraction studies. Web learners with different learning styles have different task planning preferences. These preferences lead learners to make different task plans for the same learning tasks. As reflected in our empirical results, the learning style employed in SAL task extraction can be successful in predicting web learning tasks.

Although our findings are promising, there are still some limitations of our study. First, the learning styles of a learner are highly dependent on the learning tasks. In this study, we manually analyzed and classified learning styles for UWP programs. However, methods to automatically extract learning styles may be worthy of further investigation. Second, there were limitations to the experimental data. In this study, we used the collected Northeastern University UWP coursework data as the dataset for the SAL study. This is the only dataset that contains both search activities and learning outcomes. We expect that the proposed method will be validated using large-scale datasets.

Future studies should include the following aspects: (1) developing methods for automatically extracting learning styles; (2) studying the use of latent features to predict learning outcomes to help expand the dataset and further improve the performance of SAL task extraction; (3) considering the time dimension to predict SAL tasks in future studies; (4) developing task-oriented personalized recommendation systems for web learners; (5) exploring the optimization of the learning path based on SAL tasks. One of our goals is to adopt this in recommending systems for university courses, and we will further explore the SAL path on the basis of the obtained results.

Author Contributions: Conceptualization, P.L. and B.Z.; Data curation, P.L. and Y.Z.; Formal analysis, P.L., B.Z. and Y.Z.; Funding acquisition, B.Z.; Investigation, P.L.; Methodology, P.L. and Y.Z.; Resources, P.L.; Software, P.L.; Validation, P.L., B.Z. and Y.Z.; Writing—original draft, P.L.; Writing—review & editing, P.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Key Project of the National Natural Science Foundation of China: U1908212.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References
2. Vakkari, P. Searching as learning: A systematization based on literature. J. Inf. Sci. 2016, 42, 7–18. [CrossRef]


20. Prouo-Rios, V.; González-Ibáñez, R. Dataset of Search Results Organized as Learning Paths Recommended by Experts to Support Search as Learning. Data 2020, 5, 92. [CrossRef]


47. Li, C.; Yang, Y.; Jing, Y. Formulation of teaching strategies for graduation internship based on the experiential learning styles of nursing undergraduates: A non-randomized controlled trial. BMC Med. Educ. 2022, 22, 153. [CrossRef]