Scientific Journal Impact Factor (SJIF): 1.711

e-ISSN: 2349-9745
p-ISSN: 2393-8161

International Journal of Modern Trends in Engineering and Research
www.ijmter.com

Secure Protection in Customized Web Search

Dr.J.Bhuvana[1], Ms.V.Manimekalai[2], Mr.P.Dhanasekar[3]
Associate Professor[1], Assistant Professor[2], Assistant Professor[3]
mkalai55@gmail.com[2]

***ABSTRACT*:** The web search engine has long become the most important portal for ordinary people looking for useful information on the web. However, users might experience failure when search engines return irrelevant results that do not meet their real intentions. Such irrelevance is largely due to the enormous variety of users' contexts and backgrounds, as well as the ambiguity of texts. Personalized web search (PWS) is a general category of search techniques aiming at providing better search results, which are tailored for individual user needs. As the expense, user information has to be collected and analyzed to figure out the user intention behind the issued query. The solutions to PWS can generally be categorized into two types, namely click-log-based methods and profile-based ones. The click-log based methods are straightforward— they simply impose bias to clicked pages in the user's query history. Although this strategy has been demonstrated to perform consistently and considerably well, it can only work on repeated queries from the same user, which is a strong limitation confining its applicability. In contrast, profile-based methods improve the search experience with complicated user-interest models generated from user profiling techniques. Profile-based methods can be potentially effective for almost all sorts of queries, but are reported to be unstable under some circumstances.

Keywords – PWS, Secure protection, Customized web search, utility, risk, profile

I. INTRODUCTION

The web search engine has long become the most important portal for ordinary people looking for useful information on the web. However, users might experience failure when search engines return irrelevant results that do not meet their real intentions. Such irrelevance is largely due to the enormous variety of users' contexts and backgrounds, as well as the ambiguity of texts. Personalized web search (PWS) is a general category of search techniques aiming at providing better search results, which are tailored for individual user needs. As the expense, user information has to be collected and analyzed to figure out the user intention behind the issued query.

The solutions to PWS can generally be categorized into two types, namely click-log-based methods and profile-based ones. The click-log based methods are straightforward— they simply impose bias to clicked pages in the user's query history. Although this strategy has been demonstrated to perform consistently and considerably well [1]. Although there are pros and cons for both types of PWS techniques, the profile-based PWS has demonstrated more effectiveness in improving the quality of web search recently, with increasing usage of personal and behaviour information to profile its users, which is usually gathered implicitly from query history [2], [3], [4], browsing history [5], [6], click-through data [7], [8], [1] bookmarks [9], user documents [2], [10], and so forth. Unfortunately, such implicitly collected personal data can easily reveal a gamut of user's private life. Privacy issues rising from the lack of protection for such data, for instance the AOL query logs scandal not only raise panic among individual users, but also dampen the data-publisher's enthusiasm in offering personalized service [11].

1.1 Motivations

To protect user privacy in profile-based PWS, researchers have to consider two contradicting effects during the search process. On the one hand, they attempt to improve the search quality with the personalization utility of the user profile. On the other hand, they need to hide the privacy contents existing in the user

profile to place the privacy risk under control. A few previous studies [10], [12] suggest that people are willing to compromise privacy if the personalization by supplying user profile to the search engine yields better search quality. In an ideal case, significant gain can be obtained by personalization at the expense of only a small (and less-sensitive) portion of the user profile, namely a generalized profile. Thus, user privacy can be protected without compromising the personalized search quality. In general, there are tradeoffs between the search quality and the level of privacy protection achieved from generalization.

Unfortunately, the previous works of privacy preserving PWS are far from optimal. The problems with the existing methods are explained in the following observations:

1. The existing profile-based PWS do not support runtime profiling. A user profile is typically generalized for only once offline, and used to personalize all queries from a same user indiscriminatingly. Such "one profile fits all" strategy certainly has drawbacks given the variety of queries. One evidence reported in [1] is that profile-based personalization may not even help to improve the search quality for some ad hoc queries, though exposing user profile to a server has put the user's privacy at risk. A better approach method is,

a. whether to personalize the query (by exposing the profile) and

b. what to expose in the user profile at runtime.

To the best of our knowledge, no previous work has supported such feature.

2. The existing methods do not take into account the customization of privacy requirements. This probably makes some user privacy to be overprotected while others insufficiently protected. For example, in [10], all the sensitive topics are detected using an absolute metric called surprisal based on the information theory, assuming that the interests with less user document support are more sensitive. However, this assumption can be doubted with a simple counterexample: If a user has a large number of documents about "sex," the surprisal of this topic may lead to a conclusion that "sex" is very general and not sensitive, despite the truth which is opposite. Unfortunately, few prior works can effectively address individual privacy needs during the generalization.

3. Many personalization techniques require iterative user interactions when creating personalized search results.

They usually refine the search results with some metrics which require multiple user interactions, such as rank scoring [13], average rank [8], and so on. This paradigm is, however, infeasible for runtime profiling, as it will not only pose too much risk of privacy breach, but also demand prohibitive processing time for profiling. Thus, we need predictive metrics to measure the search quality and breach risk after personalization, without incurring iterative user interaction.

II. RELATED WORKS

In this section, we overview the related works. We focus on the literature of profile-based personalization and privacy protection in PWS system.

2.1 Profile-Based Personalization

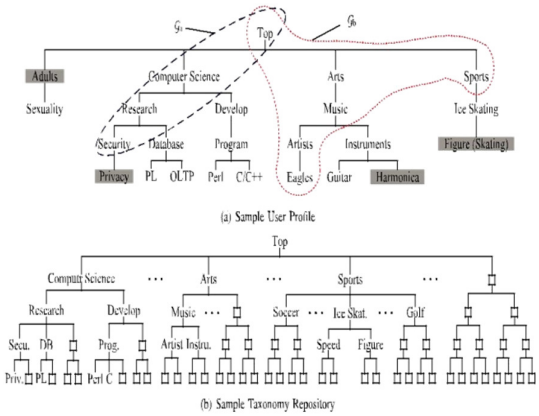
Previous works on profile-based PWS mainly focus on improving the search utility. The basic idea of these works is to tailor the search results by referring to, often implicitly, a user profile that reveals an individual information goal.

. To reduce the human involvement in performance measuring, researchers also propose other metrics of personalized web search that rely on clicking decisions, including Average Precision (AP) [10], Rank Scoring [13], and Average Rank [3], [8]. We use the Average Precision metric, proposed by Dou et al. [1], to measure the effectiveness of the personalization in UPS. Meanwhile, our work is distinguished from previous studies as it also proposes two predictive metrics, namely personalization utility and privacy risk, on a profile instance without requesting for user feedback.

. One main limitation in this work is that it builds the user profile as a finite set of attributes, and the probabilistic model is trained through predefined frequent queries. These assumptions are impractical in the context of PWS. Xu et al. [10] proposed a privacy protection solution for PWS based on hierarchical profiles. Using a user-specified threshold, a generalized profile is obtained in effect as a rooted sub tree of the complete profile. Unfortunately, this work does not address the query utility, which is crucial for the service quality of PWS.

III. USER PROFILE

Consistent with many previous works in personalized web services, each user profile in UPS adopts a hierarchical structure.



3.1 Attack Model

Our work aims at providing protection against a typical model of privacy attack, namely eavesdropping. As shown in Fig. 3, to corrupt Alice's privacy, the eavesdropper Eve successfully intercepts the communication between Alice and the PWS-server via some measures, such as man-in-the-middle attack, invading the server, and so on. Consequently, whenever Alice issues a query q, the entire copy of q together with a runtime profile G will be captured by Eve. Based on G, Eve will attempt to touch the sensitive nodes of Alice by recovering the segments hidden from the original H and computing a confidence for each recovered topic, relying on the background knowledge in the publicly available taxonomy repository R.

3.2 Generalizing User Profile

Now, we exemplify the inadequacy of forbidding operation. In the sample profile in Fig. 2a, Figure is specified as a sensitive node. Thus, rsbtr(S, H) only releases its parent Ice Skating. Unfortunately, an adversary can recover the sub tree of Ice Skating relying on the repository shown in Fig. 2b, where Figure is a main branch of Ice Skating besides Speed. If the probability of touching both branches is equal, the adversary can have 50 percent confidence on Figure. This may lead to high privacy risk if sen(Figure) is high. A safer solution would remove node Ice Skating in such case for privacy protection. In contrast, it might be unnecessary to remove sensitive nodes with low sensitivity. Therefore, simply forbidding the

sensitive topics does not protect the user's privacy needs precisely.

To address the problem with forbidding, we propose a technique, which detects and removes a set of nodes X from H, such that the privacy risk introduced by exposing G = rsbtr(X, H) is always under control. Set X is typically different from S. For clarity of description, we assume that all the sub trees of H rooted at the nodes in X do not overlap each other. This process is called generalization, and the output G is a generalized profile.

The generalization technique can seemingly be con-ducted during offline processing without involving user queries. However, it is impractical to perform offline generalization due to two reasons:

1. The output from offline generalization may contain many topic branches, which are irrelevant to a query. A more flexible solution requires online generalization, which depends on the queries. Online generalization not only avoids unnecessary privacy disclosure, but also removes noisy topics that are irrelevant to the current query.
   For example, given a query q_a = "K-Anonymity," which is a privacy protection technique used in data publishing, a desirable result of online generalization might be G_a, surrounded by the dashed ellipse in Fig. 2a. For comparison, if the query is q_b = "Eagles," the generalized profile would better become G_b contained in the dotted curve, which includes two possible intentions (one being a rock band and the other being an American football team Philadelphia Eagles). The node sets to be removed are X_a={ Adults, Privacy, Database, Develop, Arts, Sports}, and X_b = {Adults, Computer Science, Instrument, Ice Skating}, respectively.

2. It is important to monitor the personalization utility during the generalization. Using the running example, profiles G_a and G_b might be generalized to smaller rooted sub trees. However, overgeneralization may cause ambiguity in the personalization, and eventually lead to poor search results. Monitoring the utility would be possible only if we perform the generalization at runtime.

IV. UPS PROCEDURES

Generally, the offline phase constructs the original user profile and then performs privacy requirement customization according to user-specified topic sensitivity. The subsequent online phase finds the Optimal €-Risk Generalization solution in the search space determined by the customized user profile.
The cost layer defines for each node t € H a cost value cost (t) ≥ 0, which indicates the total sensitivity at risk caused by the disclosure of t. These cost values can be computed offline from the user-specified sensitivity values of the sensitive nodes. The preference layer is computed online when a query q is issued. It contains for each node t € H a value indicating the user's query-related preference on topic t.
Specifically, each user has to undertake the following procedures in our solution:

1. offline profile construction,
2. offline privacy requirement customization,
3. online query-topic mapping, and
4. online generalization.

V. GENERALIZATION TECHNIQUES

Then, we present our method of online decision on personalization. Finally, we propose the generalization algorithms.

5.1 Metrics

5.1.1 Metric of Utility

The purpose of the utility metric is to predict the search quality (in revealing the user's intention) of the query q on a generalized profile G. The reason for not measuring the search quality directly is because search quality depends largely on the implementation of PWS search engine, which is hard to predict. In addition, it is too expensive to solicit user feedback on search results. Alternatively, we transform the utility prediction problem to the estimation of the discriminating power of a given query q on a profile G under the following assumption.

5.1.2 Metric of Privacy

The privacy risk when exposing G is defined as the total sensitivity contained in it, given in normalized form. In the worst case, the original profile is exposed, and the risk of exposing all sensitive nodes reaches its maximum, namely 1. However, if a sensitive node is pruned and its ancestor nodes are retained during the generalization, we still have to evaluate the risk of exposing the ancestors.

Given a generalized profile G, the un normalized risk of exposing it is recursively given by

$$\text{Risk}(t, G) = \begin{cases} \text{cost } t & \text{if t is leaf,} \\ \sum_{t' \in c(t,G)} \text{Risk}(t', G) & \text{otherwise} \end{cases}$$

However, in some cases, the cost of a nonleaf node might even be greater than the total risk aggregated from its children. For instance, in the profile G_b (Fig. 2a), the cost of Music is greater than that of Artist since Music has sensitivity propagation from its sensitive descendent Harmonica. Therefore, (6) might underestimate the real risk. So we amend the equation for non leaf node as

$$\text{Risk}(t, G) = \max((\text{cost}(t), \sum_{t' \in c(t,G)} \text{Risk}(t', G))$$

Then, the normalized risk can be obtained by dividing the un normalized risk of the root node with the total sensitivity in H, namely

$$\text{risk}(q, G) = \frac{\text{Risk}(root, G)}{\sum_{s \in S} \text{sen}(s)}$$

We can see that risk(q, G) is always in the interval [0,1].

The benefits of making the above runtime decision are twofold:

1. It enhances the stability of the search quality,
2. It avoids the unnecessary exposure of the user profile.

We start by introducing a brute-force optimal algorithm, which is proven to be NP-hard. Then, we propose two greedy algorithms, namely the GreedyDP and GreedyIL.

5.2 The Brute-Force Algorithm

The brute-force algorithm exhausts all possible rooted sub trees of a given seed profile to find the optimal generalization. The privacy requirements are respected during the exhaustion. The sub tree with the optimal utility is chosen as the result. Although the seed profile G_0 is significantly smaller than H, the exponential computational complexity of brute-force algorithm is still unacceptable.

Theorem 1. The δ-RPG problem (Problem 1) is N P-hard.

5.3 The GreedyDP Algorithm

Given the complexity of our problem, a more practical solution would be a near-optimal greedy algorithm. As preliminary, we introduce an operator → called prune-leaf, which indicates the removal of a leaf topic t from a profile. Formally, we denote by $G_i \rightarrow G_{i+1}$ the process of pruning leaf t from G_i to obtain G_{i+1}. Obviously, the optimal profile G € can be generated with a finite-length transitive closure of prune-leaf.

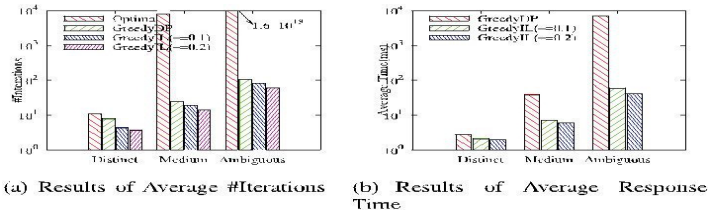
5. IMPLEMENTATION ISSUES

This section presents our solutions to some of the open problems in the UPS processing. We start by introducing an inverted-indexing mechanism for computing the query-topic relevance. Then, we discuss how the topic for each document d 2 D is detected (Offline-1) relying on this index.

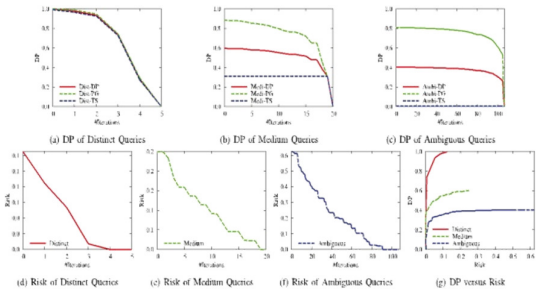
VI. EXPERIMENTAL RESULTS

In the first experiment, we study the detailed results of the metrics in each iteration of the proposed algorithms. Second, we look at the effectiveness of the proposed query-topic mapping.

The advantage of GreedyIL over GreedyDP is more obvious in terms of response time, as Fig. 6b shows. This is because GreedyDP requires much more recomputation of DP, which incurs lots of logarithmic operations. The problem worsens as the query becomes more ambiguous.



(a) Results of Average #Iterations (b) Results of Average Response Time

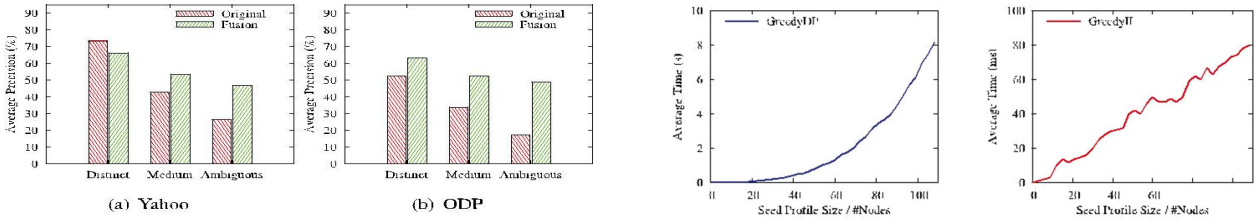


7.4 Scalability of Generalization Algorithms

We study the scalability of the proposed algorithms by varying 1) the seed profile size (i.e., number of nodes), and 2) the data set size (i.e., number of queries). For each possible seed profile size (ranging from 1 to 108), we randomly choose 100 queries from the AOL query log, and take their respective R(q) as their seed profiles. All leaf nodes in a same seed profile are given equal user preference. These queries are then processed using the GreedyDP and GreedyIL algorithms. For fair comparison, we set the privacy threshold € = 0 for GreedyIL to make it always run the same number of iterations as GreedyDP does. Fig. 7 shows the average response time of the two algorithms while varying the seed profile size. It can be seen that the cost

of GreedyDP grows exponentially, and exceeds 8 seconds when the profile contains more than 100 nodes. However, GreedyIL displays near-linear scalability, and significantly outperforms GreedyDP.

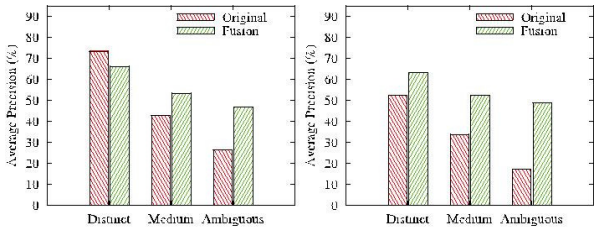
Fig. 8 illustrates the results of data sets containing different numbers of queries (from 1,000 to 100,000 queries). Apparently both algorithms have linear scalability by the data set size. For the largest data set containing 100,000 queries, it took GreedyDP 84 hours to complete all queries while GreedyIL less than 150 minutes.



(a) Yahoo (b) ODP (a) Results of GreedyDP (b) Results of GreedyIL

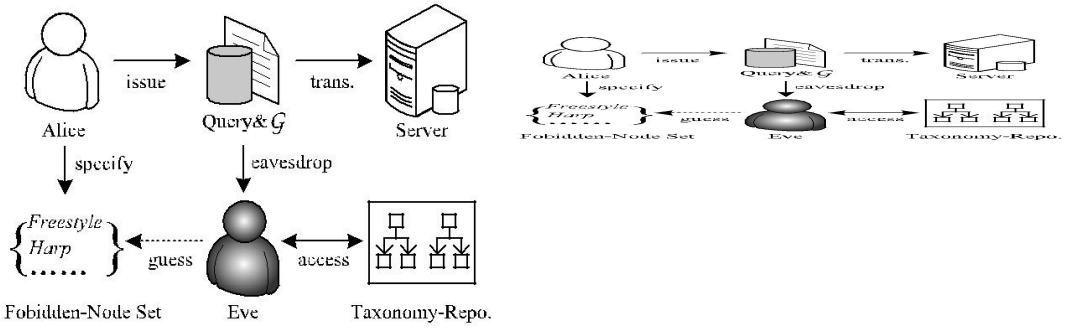
7.5 Effective Analysis of Personalization

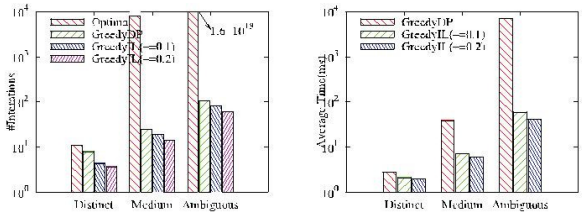
In this experiment, we evaluate the real search quality on commercial search engines using our UPS framework. The search results is re ranked with the generalized profile output by GreedyIL over 50 target users.



(a) Yahoo (b) ODP

Effectiveness of personalization on test queries.





(a) Results of Average #Iterations (b) Results of Average Response Time

VII. CONCLUSIONS

This paper presented a client-side privacy protection framework called UPS for personalized web search. UPS could potentially be adopted by any PWS that captures user profiles in a hierarchical taxonomy. The framework allowed users to specify customized privacy requirements via the hierarchical profiles. In addition, UPS also performed online generalization on user profiles to protect the personal privacy without compromising the search quality. We proposed two greedy algorithms, namely GreedyDP and GreedyIL, for the online generalization. Our experimental results revealed that UPS could achieve quality search results while preserving user's customized privacy requirements. The results also confirmed the effectiveness and efficiency of our solution.

For future work, we will try to resist adversaries with broader background knowledge, such as richer relationship among topics (e.g., exclusiveness, sequentiality, and so on), or capability to capture a series of queries (relaxing the second constraint of the adversary in Section 3.3) from the victim. We will also seek more sophisticated method to build the user profile, and better metrics to predict the performance (especially the utility) of UPS.

REFERENCES

[1] Z. Dou, R. Song, and J.-R. Wen, "A Large-Scale Evaluation and Analysis of Personalized Search Strategies," Proc. Int'l Conf. World Wide Web (WWW), pp. 581-590, 2007.
[2] J. Teevan, S.T. Dumais, and E. Horvitz, "Personalizing Search via Automated Analysis of Interests and Activities," Proc. 28th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR), pp. 449-456, 2005.
[3] M. Spertta and S. Gach, "Personalizing Search Based on User Search Histories," Proc. IEEE/WIC/ACM Int'l Conf. Web Intelligence (WI), 2005.
[4] B. Tan, X. Shen, and C. Zhai, "Mining Long-Term Search History to Improve Search Accuracy," Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD), 2006.
[5] K. Sugiyama, K. Hatano, and M. Yoshikawa, "Adaptive Web Search Based on User Profile Constructed without any Effort from Users," Proc. 13th Int'l Conf. World Wide Web (WWW), 2004.
[6] X. Shen, B. Tan, and C. Zhai, "Implicit User Modeling for Personalized Search," Proc. 14th ACM Int'l Conf. Information and Knowledge Management (CIKM), 2005.
[7] X. Shen, B. Tan, and C. Zhai, "Context-Sensitive Information Retrieval Using Implicit Feedback," Proc. 28th Ann. Int'l ACM SIGIR Conf. Research and Development Information Retrieval (SIGIR), 2005.
[8] F. Qiu and J. Cho, "Automatic Identification of User Interest for Personalized Search," Proc. 15th Int'l Conf. World Wide Web (WWW), pp. 727-736, 2006.
[9] J. Pitkow, H. Schu¨tze, T. Cass, R. Cooley, D. Turnbull, A. Edmonds, E. Adar, and T. Breuel, "Personalized Search," Comm. ACM, vol. 45, no. 9, pp. 50-55, 2002.
[10] Y. Xu, K. Wang, B. Zhang, and Z. Chen, "Privacy-Enhancing Personalized Web Search," Proc. 16th Int'l Conf. World Wide Web (WWW), pp. 591-600, 2007.
[11] K. Hafner, Researchers Yearn to Use AOL Logs, but They Hesitate, New York Times, Aug. 2006.
[12] A. Krause and E. Horvitz, "A Utility-Theoretic Approach to Privacy in Online Services," J. Artificial Intelligence Research, vol. 39, pp. 633-662, 2010.
[13] J.S. Breese, D. Heckerman, and C.M. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," Proc. 14th Conf. Uncertainty in Artificial Intelligence (UAI), pp. 43-52, 1998.
[14] P.A. Chirita, W. Nejdl, R. Paiu, and C. Kohlschu¨tter, "Using ODP Metadata to Personalize Search," Proc. 28th Ann. Int'l ACM SIGIR Conf. Research and Development Information Retrieval (SIGIR), 2005.
[15] A. Pretschner and S. Gauch, "Ontology-Based Personalized Search and Browsing," Proc. IEEE 11th Int'l Conf. Tools with Artificial Intelligence (ICTAI '99), 1999.
[16] E. Gabrilovich and S. Markovich, "Overcoming the Brittleness Bottleneck Using Wikipedia: Enhancing Text Categorization with Encyclopedic Knowledge," Proc. 21st Nat'l Conf. Artificial Intelli-gence (AAAI), 2006.
[17] K. Ramanathan, J. Giraudi, and A. Gupta, "Creating Hierarchical User Profiles Using Wikipedia," HP Labs, 2008.
[18] K. Ja¨rvelin and J. Keka¨la¨inen, "IR Evaluation Methods for Retrieving Highly Relevant Documents," Proc. 23rd Ann. Int'l ACM SIGIR Conf. Research and Development Information Retrieval (SIGIR), pp. 41-48, 2000.

