Postprint: Research on Voice Push Systems for Blind Internet Users

Authors: Xie Shuhua, Deng Zhuhui, Wang Xiangdong, Qian Yueliang, Lin Shouxun

Date: 2016-11-02T00:00:00+00:00

Abstract

Research and development of non-visual voice browsers and voice push platforms is of significant importance for enabling internet access among visually impaired individuals (including blind and low-vision persons). This paper introduces the development of non-visual voice browsers and voice push platforms, as well as the current research status of core technologies such as web page parsing and voice push, proposes a voice push platform solution based on web content parsing that employs a block-based browsing approach, and elaborates on the relevant system architecture and key technologies. Through the voice push system, visually impaired individuals will be able to conveniently and efficiently obtain network information and resources.

Full Text

Research on Voice Push Systems for Blind Users' Internet Access

Authors: Xie Shuhua, Deng Zhuhui, Wang Xiangdong, Qian Yueliang, Lin Shouxun

Abstract

Research and development of non-visual voice browsers and voice push platforms are of significant importance for enabling visually impaired individuals (blind and low-vision users) to access the Internet. This paper introduces the development of non-visual voice browsers and voice push platforms, reviews the current state of core technologies such as web page parsing and voice push, proposes a voice push platform solution based on web content parsing with block-based

browsing, and elaborates on the relevant system architecture and key technologies. With the aid of voice push systems, visually impaired individuals will be able to conveniently and efficiently obtain online information and resources.

Keywords: Non-visual voice browser; VoiceXML; Automatic web content parsing; Web page segmentation; Voice push platform

1 Introduction

With the vigorous development of Internet technology, the World Wide Web has become an indispensable source of information and communication tool in people' s daily lives. However, the Internet currently presents content primarily through visual web pages, making it impossible for blind and visually impaired individuals to effectively use the Internet like sighted people. According to the latest 2006 statistics from the China Disabled Persons' Federation, there are approximately 12.33 million blind people in China, making the need to satisfy blind users' Internet access requirements a matter of significant social importance.

With the expansion of network bandwidth and the development of technologies such as Text-to-Speech (TTS) and Automatic Speech Recognition (ASR), the implementation of non-visual voice browsers has become feasible. In recent years, research on voice push platforms and voice browsers based on speech synthesis and recognition technologies has progressed rapidly, providing blind users with a novel way to browse the Internet, obtain information, and communicate through voice, which has been welcomed by blind users. This paper provides a detailed introduction to the research and latest developments in non-visual browsers and voice push technology, proposes a solution based on web content parsing upon analyzing existing systems, discusses relevant key technologies, and presents a system implementation plan.

2.1 VoiceXML (VXML) Language

VoiceXML is a standard established by the W3C (World Wide Web Consortium) for accessing World Wide Web content and delivering interactive voice responses through voice dialogues. VoiceXML organically integrates the public telephone network, voice processing technology, and the Internet. It is a domain-specific language that defines a series of voice application concepts, elements, and their corresponding operations, enabling the definition of voice interaction processes with computers based on played audio files, output text-to-speech, recorded and recognized speech, and received DTMF tones.

VoiceXML aims to apply the vast amount of information already existing on the World Wide Web through interactive voice interfaces, while also seeking to liberate developers from low-level programming and resource handling tasks. Applying VoiceXML to non-visual browsers can conveniently handle voice interactions with users, facilitating information access for visually impaired individuals.

2.2 Non-Visual Browsers

Early non-visual browsers were primarily screen reading software. Influential foreign products include JAWS[1] and IBM Home Page Reader[2], while domestic products include Sunshine Screen Reader, Yongde Screen Reader, and Chenguang Screen Reader. Such software uses speech synthesis technology to read aloud the text content displayed on the screen, enabling blind users to access web content through hearing. The primary problem with screen reading software is that these programs perform almost no analysis or processing of web pages, instead reading all page content directly in its original order. When web pages contain substantial content, users find it difficult to quickly locate information of interest, as they must listen to large amounts of irrelevant content before making a selection.

To address the time-consuming and inefficient nature of screen reading software, recent research on non-visual browsers has gradually shifted toward analyzing and transforming web page structures to obtain pages more suitable for blind users, as well as improving the efficiency of locating and selecting web content through voice and other interactive means. Numerous researchers both domestically and internationally have conducted relevant research and developed various application systems. Representative examples include the HearSay project from the United States [3][4] and related research from IBM Japan [5]. The HearSay[3] system employs VoiceXML to handle user-system interactions and automatically converts HTML web pages into VoiceXML voice pages. To solve the efficiency problems caused by excessive page content, the system divides HTML page content into larger blocks, allowing blind users to skip between blocks through shortcut keys to quickly find information of interest. IBM Japan has adopted a similar approach [5], dividing web pages into several groups to reduce the number of items read aloud at one time. This system uses dynamic matching to batch-process entire websites, with the disadvantage of requiring offline processing and long processing times. Some non-visual browsers adopt a voice push approach, such as the WebAnyWhere[6] voice push software developed by the University of Washington, which enables blind users to conveniently use screen readers on any computer via the Internet. Blind users need not install any software; they simply access the WebAnyWhere page system through a standard browser, open the desired web page within this system, and the system will convert the page content to speech in the manner of screen reading software, pushing the voice to the client. It also supports simple shortcut key operations, greatly facilitating web browsing for blind and visually impaired individuals. However, the drawback of this voice push system is that network transmission of voice streams introduces network latency, resulting in severe delays when accessing complex web pages.

2.3 Automatic Web Content Parsing Technology

2.3.1 Related Research on Web Content Extraction Web content extraction aims to discover useful information in web pages. For instance, in

news pages, the useful information includes news titles and main text, while navigation bars and advertisement sections are considered noise. Content extraction has found applications in information retrieval, web data mining, web adaptation, knowledge acquisition, and other areas.

Current web content extraction methods fall into two main categories. The first is rule-based methods, which primarily utilize comprehensive web page information to formulate several heuristic rules for content extraction. A typical representative is the Crunch[7] system proposed by Gupta et al. This system establishes corresponding judgment rules and a series of filters for common noise information in web pages, such as advertisements and link lists, to filter out noise and indirectly achieve content extraction. The second category is automatic classification-based methods. For example, Cai-Nicolas Ziegler[8] transforms web content extraction into a classification problem by first dividing web pages into multiple sub-blocks based on tag types, then extracting features related to semantic and structural information from each block, and determining whether a block is a content block based on these features and corresponding thresholds. In determining feature thresholds, a Particle Swarm Optimizer (PSO) is used to select optimal thresholds, achieving a fully automated processing procedure. Gibson[9] converts the content extraction process into a sequence labeling problem in the natural language processing domain and implements web content classification and extraction using Conditional Random Fields (CRF).

2.3.2 Related Research on Web Page Segmentation Web page segmentation aims to discover the content structure of web pages and mine the semantic hierarchical relationships within them. In recent years, due to the rapid development of information retrieval technology and mobile handheld devices, web page segmentation technology has attracted extensive attention from scholars, who have proposed various distinctive segmentation algorithms for specific applications.

Early web page segmentation methods primarily targeted simple HTML structural tags. For example, Kaasinen et al. [10] mainly utilized tags such as `<P>`, `<TABLE>`, and `<UL>` for page segmentation. Subsequently, Deng Cai [11] et al. proposed the VIsual-based Page Segmentation (VIPS) algorithm, which has been successfully applied in information retrieval and has received widespread attention. This method primarily leverages the visual information displayed by web pages, achieving top-down segmentation through the discovery of basic blocks and separators. In addition to utilizing visual information and the DOM tree, some web page segmentation algorithms consider introducing semantic information. For instance, D. Embley [12] addresses domain-specific web page segmentation problems, using domain-specific knowledge to assist the segmentation process.

2.4 Voice Push-Related Technologies

Voice push refers to the delivery of artificially recorded or computer-synthesized speech to fixed-line or mobile phone users through the telephone network, or to network clients via the Internet. The architecture of current VoiceXML-based voice push systems is relatively mature; therefore, research hotspots focus on improving user convenience and comfort, currently concentrating primarily on voice caching and prefetching technologies.

Prefetching and caching aim to reduce the time delay from when a user issues a request to when voice is delivered to the user. Caching refers to the strategy of storing certain data in a local cache area so that when users access the same voice resource again, it can be retrieved directly from the cache to avoid transmission delays. A commonly used caching strategy is multi-level caching, which establishes cache areas at both the server and client sides, simultaneously caching recently accessed voice resources in both locations. Due to the limited size of cache areas, reasonable and efficient cache replacement algorithms are needed to update cached resources. Current cache replacement algorithms fall into three main categories: 1) Locality-based algorithms: These assume that recently accessed resources are likely to be accessed again in the near future, with the most famous being the LRU (Least Recently Used) algorithm, which always replaces the resource that has not been accessed for the longest time. 2) Frequency-based algorithms: Access frequency is a significant parameter affecting the utility function used in caching. The most commonly used LFU (Least Frequently Used) algorithm always evicts the resource with the lowest access frequency. 3) Feature-based algorithms: These replacement algorithms use key features of cached resources as conditions for or direct inputs to the utility function, with common features including resource size, resource retrieval delay, and number of resource replacements.

Although caching technology can improve system performance, relevant studies[13] indicate that regardless of the caching scheme employed, the maximum cache hit rate typically does not exceed 40%-50%. Therefore, to further improve cache hit rates, prefetching technology has been introduced. Its main idea is to predict resources that users may access in the near future and retrieve them from the server in advance. Current prefetching techniques include: 1) User access sequence-based prediction: For example, building Markov chains. Reference [14] mines user access logs from servers to obtain user access sequences and build Markov chains, matches these sequences with current access sequences, and then prefetches other pages in the matching sequence. 2) Access probability-based prediction: These algorithms assume that user access follows certain patterns. Unlike sequence-based prediction algorithms, these focus on prefetching the next resource. Reference [15] defines the probability of a user accessing resource B after accessing resource A as $P(B|A) = N(A,B) / N(A)$, where $N(A)$ is the counter for resource A, representing the number of times resource A has been accessed, and $N(A,B)$ represents the number of times resource B is accessed immediately after A. Therefore, when a user accesses A, the system can obtain

the probabilities of previously accessed subsequent resources through statistical data, thereby deciding whether prefetching is needed. 3) Hotspot-based prediction: This algorithm treats web pages and resources frequently accessed by most users as hotspots and caches these hotspot resources. 4) Link-based prediction: This approach assumes that a user' s next request often comes from links on the current page, so prefetching some links from the current page can shorten response time to user requests.

2.5 Current Research Problems

From the above review of the current state, it is evident that the main problem in research related to Internet access for blind users still lies in the difficulty of interaction between blind users and the system, manifested primarily in two aspects:

First, HTML-format web pages are not conducive to blind users' browsing. Most current systems are still based on HTML-format pages or simply convert HTML pages into VoiceXML pages or plain text. This makes it difficult for users to quickly locate content of interest as they can with visual browsing when pages are large, forcing them to listen to large amounts of irrelevant content before making a selection. Although systems like HearSay attempt to segment pages into blocks, their browsing methods are still limited to jumping to the next block without providing users with an overall page structure containing all blocks. On the other hand, while research on web content parsing exists, it primarily targets other applications such as information retrieval and has been less applied and less thoroughly explored in the context of blind user browsers.

Second, the interaction methods of voice browsers or push systems are difficult to use. Most current voice browsers are still based on web browsers, retaining many visual characteristics in their operation, such as requiring numerous complex shortcut key combinations, using menus, and mouse-operated interfaces, making it difficult for blind users to use them conveniently.

3 System Design

3.1 Design Goals and Principles

To help blind and visually impaired individuals use the Internet conveniently and efficiently, and to address the problems identified in the aforementioned research on Internet access for blind users, we have developed a server-side voice push system and corresponding client-side voice browser. The system parses web content and organizes pages into a set of sub-blocks or extracts their main content based on the characteristics of different pages, thereby helping users quickly locate and select content of interest. The voice push component of the system uses VoiceXML to handle user interactions, providing multiple interaction methods including voice commands, shortcut keys, and link selection through keywords to facilitate user use.

The system design principles are:

3.2 System Architecture Design

The voice push platform proposed in this paper supports users in accessing network information through multiple channels and methods: users can either dial into the voice platform via mobile or fixed-line telephones to access the Internet, or use computers to access the Internet through network client software. The system functional structure is shown in Figure 1 [Figure 1: see original paper].

As shown in Figure 1, the system workflow is as follows: when a user issues an access request, the VoiceXML parsing platform responds to the request and passes the network URL address that the user needs to access to the document server. The document server downloads the corresponding web page via the network based on the URL address, uses web content parsing technology to reorganize the page content (either extracting main content or organizing it into a block structure), and converts it into VoiceXML page format. Subsequently, the server-side transmits the VoiceXML page back to the VoiceXML parser platform, which parses the document and continuously modifies its own state and executes corresponding actions according to the flow within the VXML file, such as requesting speech recognition, speech synthesis, and playing voice. Based on the above system structure, the designed system architecture is shown in Figure 2 [Figure 2: see original paper].

3.3 System Function Module Division

Based on system functions, the system is divided into six modules: main flow control module, VoiceXML parsing module, speech synthesis control module, speech recognition control module, VXML document generation module, and voice resource control module, as shown in Figure 3 [Figure 3: see original paper].

The connections between the speech synthesis control module and speech synthesis server, speech recognition control module and speech recognition server, and VXML document generation module and document server are implemented through the TCP/IP protocol.

3.3.1 Main Flow Control Module The main flow control module is responsible for controlling the interaction flow between users and the system. This module accepts user calls and connections, captures user voice input and key presses, initializes various resource variables (such as speech recognition and speech synthesis resources) and environment variables, controls the initialization and shutdown of the parsing platform, launches relevant interpreter threads for user connections, and simultaneously handles returns from the parsing module.

3.3.2 VoiceXML Parsing Module The VoiceXML parsing module is primarily responsible for parsing VoiceXML documents and, during the parsing process, calling relevant module functions based on the parsing context, making it the most important module in the platform. Its main functions include: completing the parsing of VXML document tags and converting them into calls to lower-level module interfaces; completing message interaction with the voice resource control platform and controlling the execution process of parsing threads; integrating components such as speech recognition, speech synthesis, and document services, and communicating with speech recognition servers, speech synthesis servers, and document servers.

3.3.3 Speech Synthesis Control Module This module' s function is to pass text information requiring synthesis from VXML document parsing to the speech synthesis server, obtain synthesized speech from the synthesis server, and transmit it to the voice resource control module. The module connects to the speech synthesis server via HTTP, places text requiring synthesis in a queue, and sequentially sends text from the queue to the speech synthesis server for synthesis.

The speech synthesis server employs iFLYTEK' s InterPhonic 5.5 synthesis system, which can simultaneously provide anywhere from several to hundreds of speech synthesis engines according to demand, satisfying the system' s requirements for large-scale concurrent access.

3.3.4 Speech Recognition Control Module The speech recognition module transmits voice requiring recognition, passed back from the voice resource control module, to the speech recognition server via HTTP local area network, and returns the recognition results to the parsing module after the speech recognition server completes recognition. Simultaneously, this module also controls the recognition of telephone keypad and keyboard key presses, making it an important module that the VXML parsing module needs to call. The parsing module determines the next step in the workflow based on the recognition results.

3.3.5 VXML Document Generation Module The VXML document generation module communicates with the VXML parsing module via HTTP. The VXML parsing module passes the web page address requested by the user to the VXML document generation module, which retrieves the corresponding VXML document from the document server and passes it to the parsing module. If no corresponding document exists in the VXML document collection, the module downloads the relevant web page from the Internet based on the URL address and uses automatic web page parsing technology to automatically convert the HTML page into a VXML page for transmission to the parsing module.

This module employs caching and prefetching strategies. It caches some VXML documents that users have accessed in the document server, while simultane-

ously prefetching some HTML pages from the Internet and converting them into VXML documents according to certain prefetching strategies.

3.3.6 Voice Resource Control Module This module is primarily responsible for the acquisition and allocation of voice resources, including transmitting voice input from users via network or telephone to the speech recognition server, and outputting voice resources returned from the speech synthesis server to the telephone end through voice cards or transmitting them to network clients via the network. This module also employs caching and prefetching strategies.

4 Key Technologies

4.1 Web Content Parsing Technology

Since voice output is not as easily and quickly received as visual information in terms of presentation form, it can easily cause information overload during browsing. Therefore, it is essential to perform reasonable and effective analysis and organization of web page content before presenting it to users. The VXML document generation module is responsible for analyzing the HTML web page content requested by users and converting the analysis results into VXML documents. This paper proposes a web page preprocessing architecture based on page types for web content parsing, emphasizing the concept of page types and performing appropriate processing according to their respective characteristics to adapt to human-computer interaction needs for blind users and improve user experience.

The entire web page preprocessing workflow is shown in Figure 4 [Figure 4: see original paper]: First, the system determines whether the page is topic-type, then either extracts the main content block or converts HTML to VoiceXML, resulting in a VoiceXML page.

First, we introduce the concept of page types: Based on our daily Internet browsing experience, web pages can be roughly divided into two categories. One is topic pages, where there exists a clear main content block in the page, with the remaining content serving mainly as auxiliary or noise information. Typical representatives of this category are news pages, whose primary function is to deliver the content of the main content block to users. The other category is non-topic pages, which can also be called directory pages, with relatively flat content structure and no obvious main content block. Typical representatives of this category are the homepages of major websites, whose primary function is navigation, similar to a directory.

The browsing processes for these two types of pages differ significantly for most people. For topic pages, users are generally only concerned with the main content, with other information being secondary. For directory pages, there are often block structures where related content forms a small block that becomes the basic semantic and visual unit during user browsing.

Given the substantial differences in content characteristics and user browsing habits between topic and non-topic pages, for a new web page, the system proposed in this paper first determines its page type as preparation for subsequent targeted processing. To this end, we propose a heuristic rule-based page classification algorithm that comprehensively determines whether a page is a topic page using rules formed from multiple aspects of information, including link text ratio information, tag structure information, and spatial position information of web page elements.

After determining the page type, for topic pages, to highlight the main content, the system performs content extraction on such pages, dividing page content into main content and non-main content parts. To obtain as complete main content information as possible, we adopt a content extraction approach similar to that used by Gupta[7]. The main idea is not to directly target the main content of the page but to indirectly achieve main content extraction through noise removal. Since noise information in web pages often has very obvious characteristics, with the most typical being a relatively large proportion of link text, the link text ratio can be used in conjunction with web page tag structure to remove most web page noise information, and the determination conditions for noise information can be controlled to ensure the completeness of main content as much as possible.

For non-topic pages, the system performs block processing to increase the content granularity of the page and facilitate user jumping between page blocks, enabling more efficient web page access patterns. Considering the diversity of web page content and structure, using a single-layer block list structure to organize page content structure is simpler than using a multi-layer tree structure, and user interaction is also more direct and straightforward. During block processing of web pages, judgment is primarily based on tag structure information and spatial position information of the page. The entire block processing can be divided into two phases: The first phase performs a bottom-up post-order traversal of the page' s tag structure tree to calculate tag attributes required for page segmentation. The second phase performs a top-down pre-order traversal of the page' s tag structure tree, using the relevant attributes obtained in the first phase to perform tag node splitting and merging operations, thereby obtaining the final page segmentation result.

Finally, based on the differences in user browsing habits between topic and non-topic pages, the system proposed in this paper designs separate conversion templates for the two page types to implement HTML-to-VoiceXML conversion, supporting universal voice applications for both computer and telephone terminals. The conversion templates define specific user interaction patterns: First, the template alerts users to the current page type. Then, if it is a topic page, the main content block is presented to the user first; if it is a non-topic page, general information after page blocking is provided first, such as the total number of blocks and the initial text information of each block. Subsequently, users can access each page block through block numbers. The system uses a Dialog

Flow approach to organize and convert page content, simultaneously supporting both voice interaction mode and keypress interaction mode.

To verify the effectiveness of the web page preprocessing process, we conducted a series of experiments using data from the SEWM2008 Chinese information retrieval evaluation organized by Peking University' s Network Laboratory for page type judgment and content extraction experiments on topic pages. In the page type judgment experiment, the system proposed in this paper achieved an accuracy of 90.14%, a recall rate of 92.75%, and a comprehensive F-value of 91.42%, with both accuracy and F-value exceeding the best results of all participating teams in SEWM2008. In the content extraction experiment for topic pages, the system achieved an overall accuracy of 90.67%, an overall recall rate of 94.37%, and an overall F-value of 90.70%, with the recall rate exceeding the best evaluation result and the overall F-value slightly below the best result but above the second-place result. Therefore, based on the relevant data and experimental results from SEWM2008, the page type judgment algorithm and topic page content extraction algorithm proposed in this paper are highly effective. Since the evaluation of page segmentation performance is highly subjective, the system results are currently assessed primarily through subjective inspection. Figure 5 [Figure 5: see original paper] provides an example of page segmentation. It can be seen that the block boundaries and sizes are reasonable and suitable for the actual needs of blind users.

4.2 VXML Document Parsing Technology

As mentioned above, the VoiceXML document parsing module is the core of the entire platform. This paper builds the VoiceXML document parser based on OpenVXI 3.4. OpenVXI 3.4 is an open-source project that establishes a parsing framework for VoiceXML, currently supporting the complete VoiceXML 1.0 standard and most of the 2.0 standard. The voice push platform proposed in this paper is developed based on OpenVXI, including the VXI interpreter module (responsible for interpreting VoiceXML and controlling the parsing workflow, which is the core of the system), Internet interface module (enabling the platform to obtain application documents through http:// and file:// methods), XML parser (responsible for syntax analysis and providing XML DOM parsing capabilities), ECMAScript module (JavaScript, providing ECMAScript execution capabilities), logging module (responsible for reporting errors, events, and diagnostic messages to system operators and can store logs in files), and a series of application programming interfaces (APIs) that the platform needs to implement, including recognition module interface, prompt module interface, and telephone module interface, as shown in Figure 6 [Figure 6: see original paper].

4.3 Human-Computer Interaction Methods

In terms of human-computer interaction, VoiceXML session language supports controlling web page browsing through voice command recognition, as well as through Dual Tone Multi-Frequency (DTMF) keypad and keyboard key opera-

tions. To facilitate efficient interaction between blind users and the voice push platform using computers and mobile terminals, and to improve interaction efficiency, the platform proposed in this paper primarily adopts the following strategies:

First, shortcut keys are designated based on the web page segmentation strategy. Page segmentation makes user browsing faster and more convenient, and this convenience is achieved through shortcut key operations, such as jumping between blocks and jumping between items within a block. Therefore, the platform designs reasonable shortcut keys according to the needs and operation characteristics of blind users to improve interaction efficiency. To facilitate subsequent upgrades and user customization, VXML template technology is adopted in implementing shortcut keys, so that changes to shortcut keys do not require modifying system programs but only need to modify VXML script files, greatly improving system flexibility.

Second, personalized interaction modes are developed based on users' interests and preferences. For example, users can modify the voice browsing order of web pages and place news of interest in the foremost position.

Third, user operation processes are recorded. Thus, when users make the same request in the future, the system will automatically help users perform certain specific operations, such as logging into forums, checking emails, and online shopping.

5 Conclusion and Future Work

With the rapid development of voice technology, voice-based network applications have become a research hotspot. From solving Internet access for blind users to voice value-added services in call centers, and from voice browsers to voice push platforms, various design and implementation solutions have emerged for different applications. This paper provides a detailed introduction to the development and current state of non-visual voice browsers, focuses on research and progress in automatic web content parsing and voice push-related technologies, and implements a voice push platform for blind users' Internet access based on existing technologies. The platform uses web content parsing technology to reorganize web content into structures more suitable for voice browsers and adopts multiple interaction methods, greatly facilitating use by blind users. In future work, we will further investigate the habits and needs of blind users in Internet access, improve system interaction methods, and continue in-depth research on web page segmentation and voice push caching and prefetching technologies to further enhance system performance through experiments and testing.

References

[1] http://www.freedomscientific.com/

[2] C. Asakawa and T. Itoh. User interface of a homepage reader. In ASSETS, 1998.

[3] I. Ramakrishnan, A. Stent, and G. Yang. Hearsay: Enabling audio browsing on hypertext content. In Intl. World Wide Web Conf. (WWW), 2004.

[4] Jalal Mahmud, Yevgen Borodin, I.V. Ramakrishnan. CSurf: A Context-Driven Non-Visual WebBrowser. In Intl. World Wide Web Conf. (WWW), 2007.

[5] Takagi, H. and Asakawa, C., "Web Content Transcoding For Voice Output" , Technology And Persons With Disabilities Conference, 2002.

[6] Bigham et al. WebAnywhere: A Screen Reader On-the-Go. W4A 2008.

[7] S. Gupta, G. Kaiser, D. Neistadt, and P. Grimm. DOM-based content extraction of HTML documents. In WWW ' 03: Proceedings of the 12th international conference on World Wide Web, pages 207–214, New York, NY, USA, 2003. ACM.

[8] Ziegler, C. & Skubacz, M. Content Extraction from News Pages Using Particle Swarm Optimization on Linguistic and Structural Features WI ' 07: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, IEEE Computer Society, 2007, 242-249

[9] Gibson, J.; Wellner, B. & Lubar, S. Adaptive web-page content identification WIDM '07: Proceedings of the 9th annual ACM international workshop on Web information and data management, ACM, 2007, 105-112

[10] Kaasinen, E., Aaltonen, M., Kolari, J., Melakoski, S., and Laakko, T., Two Approaches to Bringing Internet Services to WAP Devices, In Proceedings of 9th International World-Wide Web Conference, 2000, pp. 231-246.

[11] Cai, D., Yu, S., Wen, J.-R. and Ma, W.-Y., VIPS: a vision-based page segmentation algorithm, Microsoft Technical Report, MSR-TR-2003-79, 2003.

[12] D. Embley and L. Xu. Record location and reconfiguration in unstructured multiple-record web documents. In WebDB, 2000.

[13] DOUGLIS F, FELDMANN A, KRISHNAMURTHY B, et al. Rate of change and other metrics: a live study of the world-wide Web[A]. Proceedings of the 1997 USENIX Symposium on Internet Technologies and Systems[C]. Monterey, 1997. 147-158.

[14] SU Z, YANG Q. Whatnext: a prediction system for Web requests using n-gram sequence models[A]. Proceedings of the First International Conference on Web Information System and Engineering Conference[C]. Hong Kong, 2000.

[15] JIANG Z, KLEINROCK L. An adaptive network prefetch scheme[J]. IEEE Journal on Selected Areas in Communications, 1998, 16(3): 358-368.

Author Biographies

Xie Shuhua: Graduate student, Pervasive Computing Center, Institute of Computing Technology, Chinese Academy of Sciences

Deng Zhuhui: Graduate student, Pervasive Computing Center, Institute of Computing Technology, Chinese Academy of Sciences

Wang Xiangdong: Ph.D., Assistant Researcher, Pervasive Computing Center, Institute of Computing Technology, Chinese Academy of Sciences

Qian Yueliang: Senior Researcher (Professor level), Director, Pervasive Computing Center, Institute of Computing Technology, Chinese Academy of Sciences

Lin Shouxun: Ph.D., Professor, Pervasive Computing Center, Institute of Computing Technology, Chinese Academy of Sciences

Note: Figure translations are in progress. See original paper for figures.