URL-Feature-Based Hub Web Page Identification (Postprint)

Authors: Zhang Ce, Du Yuncheng, Liang Ran

Date: 2017-10-11T00:00:00+00:00

Abstract

Objective: By constructing simple data samples, this study addresses the challenge of low efficiency in traditional web page type identification methods. Methods: URL features are employed as the basis for identification. URL information is extracted to construct training and test sets, and a Support Vector Machine (SVM) is used to establish a machine learning model to enhance identification efficiency. Results: On the same dataset, the proposed method achieves an accuracy of 91.2%, outperforming other identification methods. In terms of efficiency performance, the method demonstrates an improvement of nearly 60%. Limitations: When encountering websites where URL features are not obvious or even completely contradictory, the identification accuracy decreases substantially. Conclusion: This method offers significant advantages in efficiency, and its application to collection systems can improve collection efficiency.

Full Text

A Study on Hub Page Recognition Using URL Features

Zhang Ce[1], Du Yuncheng[1],[2], Liang Ran[2] [1](Open Laboratory of TRS Software, Beijing Information Science and Technology University, Beijing 100085, China) [2](Beijing TRS Information Technology Co. Ltd., Beijing 100101, China)

Abstract

[**Objective**] This study addresses the low efficiency of traditional web page type recognition methods by constructing simple data samples. [**Methods**] URL features serve as the basis for recognition. URL information is extracted to construct training and test sets, and a Support Vector Machine (SVM) model is established to improve recognition efficiency. [**Results**] On the same dataset, this method achieves 91.2% accuracy, outperforming other recognition methods, while improving efficiency performance by nearly 60%. [**Limitations**] When

encountering websites where URL features are not obvious or even contradictory, recognition accuracy decreases substantially. [**Conclusions**] This method offers significant advantages in efficiency and can improve collection efficiency when applied to crawling systems.

Keywords: URL features; Hub pages; SVM

With the development of the Internet, the number of web pages has grown rapidly. Even with large-scale distributed web crawling systems, collecting the vast majority of important web pages across the entire network requires considerable time. Research indicates that only about 8.52% of Chinese web pages change within a month [1]. Therefore, full-collection approaches entail significant resource waste. Additionally, due to long crawling cycles, pages with high change frequency undergo multiple modifications between crawls, yet crawling systems cannot promptly capture these changes, preventing search engines from providing retrieval services for these pages. To address this problem, incremental web crawling systems have emerged.

Incremental crawling systems do not collect all discovered URLs. Instead, they estimate web page change patterns to collect only newly emerged, changed, or disappeared pages, ignoring unchanged pages. This greatly reduces crawling volume, enables rapid synchronization between web pages and search engine indexes, and provides users with more real-time retrieval services.

In incremental crawling research, web pages are typically categorized into Hub pages and Topic pages [2]. Hub pages serve as directories that guide users to relevant topic pages, providing entry points without containing specific content themselves [3]. Topic pages, by contrast, discuss specific subjects in detail. Experiments have demonstrated that many new pages are linked from Hub pages [4]. Consequently, incremental crawling systems can discover new URLs by identifying and crawling Hub pages. Thus, recognizing which pages are Hub pages becomes the primary challenge.

To address this issue, this paper proposes a Hub page recognition method based on URL features. For the first time, URL features serve as the sole basis for Hub page recognition, which will compensate for the substantial overhead of traditional Hub page recognition methods. Comparative experiments validate this approach.

Current main Hub page recognition methods include simple rule-based methods [4], multi-feature heuristic rule-based classification methods [5-6], and web content-based machine learning methods [7-9].

Simple rule-based methods analyze Hub page URL characteristics, summarize patterns, and establish simple rules—pages meeting these criteria are identified as Hub pages. Meng et al. proposed selecting website homepages and pages whose filenames contain words like "index," "class," and "default" as Hub pages [4], then crawling pages linked from these Hub pages. While this method can

collect a large portion of new pages, its recall rate for new page collection is not high due to two problems: (1) Inaccurate Hub page selection. Since filenames are human-assigned without fixed patterns, no single rule can correctly identify all Hub pages. (2) Inability to automatically recognize Hub pages. During crawling, new Hub pages cannot be discovered promptly, so link information within them remains undiscovered.

To overcome these limitations, Ali et al. proposed a multi-feature heuristic rule-based classification method using three features: non-link character count, punctuation count, and text-to-link ratio [5]. Research shows extensive differences between Hub and Topic pages in these feature values, demonstrating classification feasibility. This method calculates each feature' s probability support for Hub pages using Bayes' formula, computes comprehensive support, and compares it with a threshold to determine page type. However, this approach overly depends on threshold setting, which directly affects classification accuracy. Since different website types require different thresholds, algorithm complexity increases.

To address threshold dependency, reference [9] proposed a web content-based machine learning method that analyzes web page features through HTML parsing to establish training and test sets for Hub page recognition. While accurate, this method is inefficient and adds system overhead because it requires parsing all HTML pages and extracting features, consuming system resources and burdening the crawling system.

Building upon previous research, this paper' s URL feature-based recognition method largely solves these problems. Using URL features as samples and SVM as the machine learning approach, this method offers greater practical value compared to rule-based and content-based methods. Feature extraction is simple, efficient, and easy to implement while maintaining recognition accuracy. Moreover, since URL extraction is essential in crawling systems, using URL features as recognition criteria minimizes impact on system efficiency without adding substantial overhead.

3.1 Introduction to SVM

Support Vector Machine (SVM), developed by Vapnik et al., is a machine learning method based on statistical learning theory—specifically VC dimension theory and structural risk minimization. SVM particularly outperforms other algorithms with small sample sizes [10-12].

The fundamental concept defines an optimal linear hyperplane, reducing the search for this hyperplane to solving a convex optimization problem. Based on Mercer' s kernel expansion theorem, SVM maps samples to high-dimensional or even infinite-dimensional feature spaces through nonlinear mapping, enabling linear learning methods to solve highly nonlinear classification and regression problems in sample space. SVM offers several advantages: (1) Based on structural risk minimization, it avoids overfitting and provides strong generalization. (2) It is a theoretically grounded small-sample learning method that essentially

bypasses traditional induction-to-deduction processes, achieving efficient transductive inference from training to prediction samples. (3) The final decision function depends only on a few support vectors, making computational complexity dependent on support vector count rather than sample space dimensionality, thus avoiding the "curse of dimensionality." (4) With few support vectors determining the final result, the method captures key samples while eliminating redundancy, ensuring algorithmic simplicity and robustness.

3.2 Method Overview

Hub page recognition can be understood as a binary classification problem, where the positive class represents Hub pages and the negative class represents Topic pages. The key challenge lies in correctly distinguishing between them.

The URL feature-based method classifies pages according to URL features related to Hub pages. The process involves: analyzing obtained URLs to extract feature information and identify Hub page-related features; integrating these features into training and test sets; training an SVM model with the training set while evaluating its performance; adjusting SVM parameters based on results to determine optimal parameters and obtain the final model.

3.3 Implementation Process

[Figure 1: see original paper] illustrates the architecture of the URL feature-based Hub page recognition method, which comprises three main modules: preprocessing, feature extraction, and training/classification.

Preprocessing primarily involves URL analysis. URLs contain much information, some of which can serve as classification criteria. URL analysis aims to identify useful feature information, including URL length and whether certain strings appear. Anchor text corresponding to URLs also reflects page type to some extent and must be extracted during preprocessing. All basic data in this experiment were previously collected by a web crawler, which recorded URLs and corresponding titles as log files. The experiment extracts and analyzes these logs to obtain URL-related information, including: URL title length, URL length, date presence, filename, file type, parameter names, parameter count, directory name, directory depth, page size, and crawl depth.

Feature extraction includes feature selection and quantization. Feature selection removes low-information, unimportant features to reduce dimensionality. Feature quantization converts selected features into numerical values representing their association with Hub pages.

Through URL analysis, we identified distinguishing Hub page features: (1) URL title length (anchor text length) is generally shorter since Hub pages don' t discuss specific content. (2) URL length is shorter as Hub pages reside above Topic pages. (3) Date presence: Topic pages often contain publication dates, while Hub pages rarely do. (4) Filename: Hub page URLs are typically either

directory-only or contain "index," "class," "default," or "list." (5) File type: Hub pages with filenames are mostly ASP, JSP, ASPX, or PHP types. (6) Parameter names: Topic page URLs often contain ID parameters, while Hub pages generally have none. (7) Parameter count: Hub page URLs mostly have no parameters. (8) Directory depth: Hub pages typically reside at upper website levels. (9) Page size: Hub pages contain many links and are relatively large. (10) Crawl depth: Hub pages are generally crawled before Topic pages since they provide entry links.

Since machine learning models only classify numerical types, text types must be numerically encoded. Encoding involves analyzing different URL text values, identifying representative values, and assigning weights based on frequency statistics. In this experiment, 500 Hub pages were selected to count text value occurrences, calculate probabilities, and normalize them by multiplying by 100 to obtain reasonable feature value ranges. Specifically: (1) For filenames: 302 empty filenames (probability 0.604, value 60.4); 153 containing "class," "index," "default," or "list" (probability 0.306, value 30.6); 0 containing "article" or "content" (probability 0, value 0); 45 other cases (probability 0.09, value 9). (2) For file types: 302 empty types (probability 0.604, value 60.4); 123 containing "asp," "jsp," "aspx," or "php" (probability 0.246, value 24.6); 75 containing "shtml," "html," or "htm" (probability 0.15, value 15); 0 other cases (probability 0, value 0). (3) For parameter names: 412 empty (probability 0.824, value 82.4); 52 containing "id" (probability 0.104, value 10.4); 36 other cases (probability 0.072, value 7.2).

Training and classification converts URLs into vector space representations using LibSVM [13] for classification. LibSVM is an integrated package for fast and effective SVM pattern recognition and regression that provides source code for customization. This experiment uses Java source code from LibSVM-3.20, modifying it for parameter auto-optimization and model file saving.

The algorithm requires data in the format: `[label] [index1]:[value1] [index2]:[value2]...` where label represents the class (typically integers), index represents feature numbers (starting from 1), and value represents feature values (typically real numbers). Features with zero values can be omitted, making indices potentially discontinuous.

Data scaling is performed first. Since original data may have overly large or small ranges, svmscale rescales data to appropriate ranges (default [-1,1]) to avoid numerical difficulties when computing kernel function inner products. The RBF kernel is selected for three reasons: it maps samples to higher-dimensional space, includes linear kernels as a special case, requires fewer parameters, and performs similarly to other kernels for certain parameters. In RBF, the gamma parameter represents kernel radius and implicitly determines data distribution in the new feature space.

SVM training uses C-SVC (C-class support vector classification), which allows incomplete classification with penalty factor c. Larger c values reduce misclas-

sification but decrease generalization; smaller c values increase misclassification but improve generalization. Cross-validation selects optimal parameters c and g (gamma). The training set size is determined by testing three sets: 1,000 samples yield 80% average accuracy, while 2,000 and 3,000 samples achieve approximately 91%. To ensure conciseness, 2,000 samples are selected, achieving 91% average classification accuracy with c=32 and g=0.0625.

The modified LibSVM source code uses grid search for automatic parameter optimization, performing ten-fold cross-validation to find parameters yielding maximum average accuracy. The trained model is saved locally to avoid retraining for each prediction. For prediction, new X values are input to obtain predicted Y values from the trained model.

4.1 Validation Method

The proposed method' s feasibility is verified through comparative experiments with: (1) traditional multi-feature heuristic rule-based classification, and (2) traditional content feature-based machine learning methods. Comparison with simple URL rule-based methods was omitted because Cao Guifeng [6] already demonstrated their inferiority to multi-feature heuristic methods.

Feasibility is evaluated from both efficiency and effectiveness perspectives. Since previous studies only reported effectiveness data, this paper reimplements both comparison methods according to original procedures to obtain efficiency metrics while achieving original effectiveness levels.

4.2 Implementation of Validation Methods

(1) Multi-feature heuristic rule-based classification involves: preprocessing to remove comments, scripts, and CSS using regular expressions; calculating normalized feature values for non-link character count, punctuation count, and text-to-link ratio; computing comprehensive support for Topic pages using these features; and comparing support with a threshold to determine page type. The threshold is determined experimentally by calculating Topic page support for 500 Hub pages, finding values concentrated below 0.6 (mostly below -0.2), then testing thresholds within this range to identify the optimal value maximizing accuracy.

(2) Content feature-based machine learning involves: HTML parsing to build a DOM tree and remove irrelevant code (style, script, applet tags) while correcting erroneous tags; extracting information including page depth, update cycle, anchor text count, text count, URL count, and new URL count; selecting eight content features showing differences between page types (page depth, update cycle, anchor text count, text count, anchor-to-text ratio, URL count, new URL count, new URL ratio); and training an SVM classification model using these features.

5.1 Evaluation Metrics

Recognition effectiveness is evaluated from two aspects: efficiency (system overhead including time, memory, and CPU usage) and effectiveness (accuracy and recall). Precision represents the ratio of correctly labeled pages in the test set, reflecting classification accuracy. Recall represents the ratio of correctly labeled pages among all pages of that class, reflecting comprehensiveness. Since precision and recall are complementary, F1 score is used as the primary evaluation standard, reflecting their combined effect.

Parameters are defined in . Precision, recall, and F1 are calculated as follows:

Precision = TP / (TP + FP)
Recall = TP / (TP + FN)
F1 = 2 × Precision × Recall / (Precision + Recall)

5.2 Experimental Data and Environment

The experiment crawled pages from 50 Chinese websites: government (10), education (10), institutional (10), corporate (10), and news (10) sites. News sites, having larger page volumes, contributed 500 pages each, while other types contributed 300 pages each. To ensure diversity and simplicity while reducing redundancy, 1,000, 2,000, and 3,000 pages were selected as training sets, evenly distributed across site types with equal Hub and Topic page counts. Three training sets were constructed to determine appropriate size.

For testing, an additional 1,000 pages from 30 websites were annotated (600 Hub pages, 400 Topic pages) to ensure comparability with existing algorithms and demonstrate the proposed method' s stability. The experimental environment used Windows 7, Intel dual-core CPU, and 2GB RAM.

5.3 Experimental Results

Three experiments were conducted using: (1) URL feature-based Hub page recognition, (2) multi-feature heuristic rule-based classification, and (3) content feature-based machine learning.

shows URL feature-based results: Precision = 91.20%, Recall = 86.33%, F1 = 88.70%. Ten-fold cross-validation on training samples yielded 91% average accuracy.

shows multi-feature heuristic rule-based results. Testing thresholds from -0.2 to 0.6 revealed -0.1 as optimal, achieving Precision = 86.63%, Recall = 83.17%, F1 = 84.86%—matching Cao Guifeng' s [6] results.

shows content feature-based results: Precision = 88.73%, Recall = 90.50%, F1 = 89.61%—matching reference [9] results.

presents system overhead data for all three methods, including time consumption, memory usage, and CPU usage.

5.4 Analysis and Discussion

To verify stability, ten-fold cross-validation during training yielded 91% average accuracy, while testing on separate test data achieved 91.2% accuracy—no significant difference, proving generality and stability. Comparative results are shown in [Figure 2: see original paper].

The URL feature-based method outperforms multi-feature heuristic rule-based classification due to: (1) heuristic rules' lack of flexibility for all web pages, (2) arbitrary threshold setting, and (3) the machine learning model' s ability to discover intrinsic feature relationships with strong generalization.

The URL feature-based method shows minimal difference from content feature-based methods in effectiveness since both use similar recognition approaches with different feature objects. URL features yield higher precision because Hub page URLs exhibit distinct characteristics (short titles/lengths, no dates). However, URL variability reduces recall when URLs don' t conform to general patterns. Content features achieve higher recall because Hub pages universally contain many links with minimal text, though some Topic pages with many links and short text reduce precision. Overall, both methods show similar effectiveness but differ significantly in efficiency.

As shows, the URL feature-based method offers substantial efficiency advantages: 70% reduction in time consumption because URL feature extraction is simpler than HTML parsing; memory and CPU usage are approximately 60% of traditional methods, minimizing impact on crawling systems. Since URL extraction is already necessary during crawling, this method adds minimal overhead without affecting crawling efficiency. Therefore, the URL feature-based Hub page recognition method holds theoretical significance and practical value.

This paper proposes a Hub page recognition technique using URL features to train machine learning models for automatic recognition. Experiments demonstrate that while achieving effectiveness comparable to traditional methods, it reduces system overhead by approximately 60%. However, limitations exist: URL variability causes accuracy drops when features are unclear or contradictory, requiring combination with content features. Future research should integrate both approaches to adapt to all websites.

References

[1] Meng Tao, Yan Hongfei, Wang Jimin. Characterizing Temporal Locality in Changes of Web Documents [J]. Journal of the China Society for Scientific and Technical Information, 2005, 24(4): 398-406.

[2] Li Xiaoming, Yan Hongfei, Wang Jimin. Search Engine: Theory, Technology and System [M]. Beijing: Science Press, 2005.

[3] Cho J, Garcia-Molina H. The Evolution of the Web and Implications for an Incremental Crawler [C]. In: Proceedings of the 26th International Conference

on Very Large Data Bases, 2002.

[4] Meng T, Yan H, Wang J, et al. The Evolution of Link-attributes for Pages and Its Implications on Web Crawling [C]. In: Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence, 2004.

[5] Ali R, Beg N M S. An Overview of Web Search Evaluation Methods [J]. Computers & Electrical Engineering, 2011, 37(6): 835-848.

[6] Cao Guifeng. Design and Implement of Webpage Classify and Clean in Search Engine [D]. Wuhan: Wuhan University of Technology, 2013.

[7] Zhang X, Zhou M, Geng G, et al. A Combined Feature Selection Method for Chinese Text Categorization [C]. In: Proceedings of the 2009 International Conference on Information Engineering and Computer Science, 2009.

[8] Xie Guanghua. Research and Application of Chinese Web Page Automatic Classification [D]. Dalian: Dalian University of Technology, 2007.

[9] Wang R J, Wang D J. Web Information Acquisition by Personal Search Engine Based on SVM [J]. International Journal of Information Acquisition, 2005, 2(4): 345-352.

[10] Pang Jianfeng, Bu Dongbo, Bai Shuo. Research and Implementation of Text Categorization System Based on VSM [J]. Application Research of Computers, 2001, 18(9): 23-26.

[11] Li Liang, Liu Wanchun, Xu Quanqing, et al. A Professional Chinese Web Page Classifier Based on Support Vector Machine [J]. Computer Application, 2004, 24(4): 58-61.

[12] Zhang Xuegong. Introduction to Statistical Learning Theory and Support Vector Machines [J]. Acta Automatica Sinica, 2000, 26(1): 32-42.

[13] Chang C C, Lin C J. LIBSVM: A Library for Support Vector Machines [J]. Transactions on Intelligent Systems and Technology, 2011, 2(3): Article No.27.

[14] Jiang J, Song X, Yu N, et al. Focus: Learning to Crawl Web Forums [J]. IEEE Transactions on Knowledge and Data Engineering, 2013, 25(6): 1293-1306.

[15] Le A, Markopoulou A, Faloutsos M. PhishDef: URL Names Say It All [C]. In: Proceedings of the 30th IEEE International Conference on Computer Communications (INFOCOM), Shanghai, China. 2011.

Author Contributions: Du Yuncheng: Conceived research direction and framework, provided revision suggestions; Zhang Ce: Designed research protocol, conducted experimental design and analysis, drafted manuscript; Liang Ran: Collected basic data; Zhang Ce and Liang Ran: Revised manuscript and finalized version.

Received Date: 2015-06-25
Revised Date: 2015-08-13