A WEB LIKELY-WORD INSTANT ORGANIZER (WEBLIO)
Dynamic Hints During Knowledge Collectors Move Mouse Over a Sentence

Po-Hsun Cheng
Software Engineering Department, National Kaohsiung Normal University
62, Shenjhong Rd., Kaohsiung, 82444, Taiwan

Ying-Pei Chen, Mei-Ju Su
Graduate Institute of Electronic Engineering, National Taiwan University
1, Sec. 4, Roosevelt Road, Taipei, 10617, Taiwan

Keywords: Data management, Information extraction, Intelligent data, Semantic web.

Abstract: The more complicated web resources exist, the more professional web browsing technologies should be innovated. This paper illustrates a concept for how to extract a web page semantic content and automatically follow the cursor location to organize the likely-words from a sentence for data intelligence. Such a web browsing concept could be implemented with a couple of cross-browser techniques. We believe this concept will be popular with any other miscellaneous form in the future browsers. However, this concept will be another important step for human-computer interaction, especially, for minimizing the time expense and maintaining the likely keywords library during further web surfing utilization.

1 INTRODUCTION

When the Mosaic browser was announced in 1993, the web resources blasted in fashionableness. A statistical web site distributes the Internet usage and population statistics that essences out the usage growth rate is 205.5% from 2000 to 2008 in the world (InternetWorldStats.com, 2008). Those facts can imply that users surf in the sea of the webs and try to collect practical knowledge for further employment. Nevertheless, we conceive most users will lose their knowledge searching direction after several browsing waves in the web sea. Consequently, it will be a problem for how to instantly organize knowledge searching routes from a specific implication sentence during the web browsing stage.

Likewise, in 2003, the Web 2.0 terminology proposed by D. Dougherty to illustrate the trend which there was an apparent alteration in how people and businesses were utilizing the web and elevating web-based applications. That is to say, the web information still blows up. Then researchers found that service-centric systems are a prominent multidisciplinary paradigm concerned with software that are constructed as compositions of self-governing services (Nano and Zisman, 2007). For the moment, Vitvar said when no single service can gratify the entire goal, the composition task tries to create a plan for that goal (Vitvar et al., 2007). Therefore, we might follow the Ding's comments to assist users and software agents find relevant knowledge on the semantic web and examines the ontologies and facts that are encoded in semantic web documents (Ding et al., 2005).

Although the hindrances for supplying such an service might be moderately unyielding, nevertheless, Goth expressed that the obstacles in integrating management data across various boundaries demonstrates a mother lode of opportunity (Goth, 2007). Meanwhile, Oren recommended that we can manipulate resources depending on the source's data access permissions and capabilities (Oren et al., 2007). Accordingly, we believe these suggestions will be upsurge the accomplishment opportunity for solving affiliated problems and also intensify our certainty.

Moreover, some researchers believe that extending service oriented architectures with semantics can assist to invent service centric information systems that better conform to transformations throughout software systems' lifetime and emerging applications exploit the capability of a new breed of semantic technologies (Vitvar et al., 2007; Hendler, 2008). One of them also predicted that emerging Web 3.0 corporations are combining the web data resources, standard languages, ever-better tools, and ontologies into ap-

Cheng P., Chen Y. and Su M.
A WEB LIKELY-WORD INSTANT ORGANIZER (WEBLIO) - Dynamic Hints During Knowledge Collectors Move Mouse Over a Sentence.
DOI: 10.5220/0001843604350438
In *Proceedings of the Fifth International Conference on Web Information Systems and Technologies (WEBIST 2009)*, page
ISBN: 978-989-8111-81-4

plications that take advantage of the power of this new species of semantic technologies (Hendler, 2008). For instance, Missikoff has built a software environment that supports the construction and assessment of a domain ontology for intelligent information integration within a virtual user community (Missikoff et al., 2002).

Primarily, the further complicated web resources exist, the more experienced web browsing through technologies would be innovated. Established on the preceding listed problems, we proposed this paper to depict a web likely-word instant organizer (WebLio) for transforming and probing user's view of intentions. That is, we compose related web searching keywords instantaneously in order to reduce the searching time for the Internet users, particularly for students and other knowledge collectors. This proposed methodology declares an intelligent keyword list which relates to present cursor location, extracts the corresponding sentence from a web page, parses potential keywords, and wraps with Google search commands.

2 METHODOLOGY

The following paragraphs illustrate our WebLio methodology for instantly constructing a clever keyword list from an active web page. We also draw a figure to allude to such a successive step in Fig. 1. It comprises at least five principal processes: determine the cursor location in a web page, obtain a sentence from the cursor location, parse sentence and compose keywords, map keywords to Google search syntax, and then display tips-on-demand at cursor location.

Fundamentally, our WebLio based on the Google Web Toolkit (GWT) to implement related functionalities in order to avoid writing web applications within an error-prone process. In addition, such an emerging process will decrease the difficulties for building, reusing, and maintaining large JavaScript code bases and AJAX components.

2.1 Get Cursor Location

Preliminary of all, our methodology attempts to obtain cursor location from an active web page. The cursor location might be accompanying with mouse moving, pointer moving, or keyboard typewriting. In addition, we only intellect the user mouse moving process during browsing through the web pages. Essentially, the web browser system will be feedback mouse location with two coordination position. The World Wide Web Consortium (W3C) proposed a Simple API for XML (SAX) standard, and we emerge the event-based StAX API to obtain the mouse position. Streaming API for XML (StAX) is an API that enables you to read and write XML documents in Java. StAX is a parser independent, pure Java API founded on interfaces that can be implemented by multiple parsers. StAX was introduced in Java 6.0 and is deliberated superior to SAX and Document Object Model (DOM).

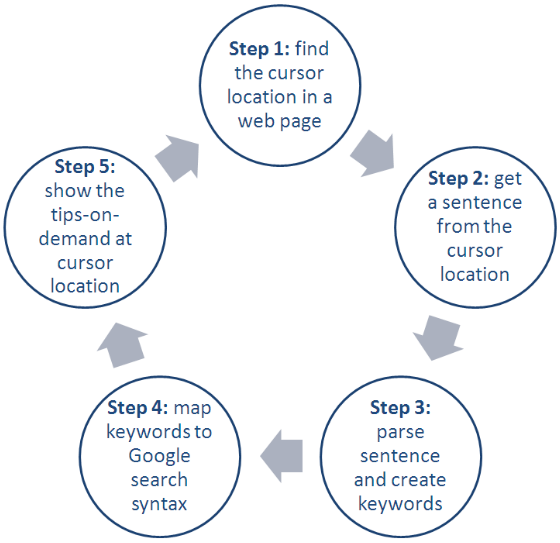
Step 1: find the cursor location in a web page
Step 2: get a sentence from the cursor location
Step 3: parse sentence and create keywords
Step 4: map keywords to Google search syntax
Step 5: show the tips-on-demand at cursor location

Figure 1: The web likely-word instant organizer (WebLio) construction methodology.

2.2 Obtain Web Sentence

That is, we can invoke methods such as getName() and getText() on the XMLStreamReader to retrieve information about the item where the cursor is currently positioned. The interface XMLStreamReader represents a cursor that is moved across an XML document from beginning to end. At any given time, the cursor always moves forward and usually only moves one item at a time.

On the other hand, the tree-based DOM technology from the W3C could let us access to all the elements on a web page, so we could refer to the cursor location and use DOM technology to locate and obtain the specific sentence which is pointed by the cursor in a specific web page. The WebLio finds and highlights the sentence element using getElementById() and className().

2.3 Parse Specific Sentence

After we obtain the specific sentence from an active web page, we utilize a self-defined parser to parse it and create an array of tokens which are extracted from a sentence. Most of the tokens in this array are single vocabulary at first, then we try to concatenate the preceding tokens and the following tokens into other keyword phrases. Such a task will be executed and also append keyword phrases at the end of the array.

Our WebLio adopts the link grammar parser application programming interface (API) (Sleater and Temperley, 1993) to handle the sentence. This implicit processing will create verbs and nouns in an English sentence and then deposit these tokens into a temporarily tiny database for next processing step. Undoubtedly, this natural language processing is a tough task and the complete success parsing possibility is depended on the parsing algorithm. Anyhow, we choose this popular open source grammar parsing algorithm as our sentence parsing basis. The advantage is that we can develop another interface to dynamic connect with different grammar parsing API's, if there are diverse grammar parsers in the world. Meanwhile, the international language switching will be more efficient.

In order to temporarily deposit the keywords in an array for processing, we utilize an open source slight database, SQLite v3.6.4, to take care of such an array. Basically, the SQLite database is an in-process software library that implements a self-included, server less, zero-configuration, and transactional SQL database engine. The SQLite is a highly deployed SQL database engine in the world and the source code is also in the public domain. Accompanied by all features qualified, the library size can be not so much than 250KB, relying on compiler optimization environments.

2.4 Likely-word Mapping

Subsequently, we use the array of keywords which is created at the preceding step to practice the keyword mapping task. That is, we endeavor to map our keywords one by one to Google search command. For instance, if there is a keyword ’human’, then we will create a uniform resource locater (URL), http://www.google.com/search?q=human, to refer to the Google search command. Correspondingly, we as well as can configure more detail search command for Google searches with the locale definition of Microsoft Internet Explorer and get the same or similar results form Google search engines.

2.5 Hints On-demand

At last, we attempt to benefit a notation container to accompany with the right mouse click to display the affiliated instant keyword list. If some of the knowledge collectors want to instant capture all of the possible likely words during mouse moving over a sentence, the system can be set for such an instant mode rather the host mode. No matter that the instant mode will detect the mouse moving and capture the possible sentence after mouse is located beyond one sentence for a second.

Generally, the host mode for our system will obtain higher performance than instant mode. After executing either mode, all of the keywords in the instant catalog can immediately open another web browser and pass the related URL. Extraordinarily, you might desire to open all of the keywords in the array list, and we as well supply such a workability to open all of the explorations which are related all of the keywords instantly by grouping.

3 RESULTS AND DISCUSSION

This document depicts a thought for how to extract a semantic web content and automatically follow the cursor position to systematize the keywords from a sentence. Such a web browsing concept will be carried out with a few cross-browser techniques. We conceive this concept will be favorite with any other miscellaneous fitness in the future browsers. Nevertheless, this concept will be another prominent stride for human-computer fundamental interaction, especially, for deprecating the time spending during web surfing.

3.1 Not a Keyword Tool External

Practically, some users might deliberate for our methodology will generate a list of keywords which is similar to some of the Keyword Tool External (KTE) tools, such as the SEOTools from SEOBook Co. The KTE is a keyword insinuation tool which was showing specific numbers for search terms instead of just color bars. Anyway, our instant web keyword generator is not belonging to the KTE tools, and it is another kind of the keyword generator which will supply the Internet users to smooth shift their focus to another search keyword which is parsed and acquired from a sentence and is located subordinate to the cursor pointer.

3.2 Security Consideration

Nonetheless, there is no important privacy thoughtfulness for us to think about it. That is, it is not so privacy for users to keep confidential from the other Internet users, in spite of; we benefit the SQLite database to affirm the keywords in the client side. We believe such a utilization of SQLite might be as well satisfactory for the majority of the Internet users.

3.3 Performance Consideration

The more hardware facilities are publicized, the higher performance for web browsing through process will be procured. This is a straightforward reasoning procedure that the web browsing operation will not be an inconvenience in the hereafter web browser with client-side high-speed hardware platform. Considering that our branch of philosophy only processes the text-based processing, it will use slighter resources than the other multimedia system. For that reason, our methodology approximately has no coincidence to become the dealing with performance bottleneck during the web browsing stage.

On the other hand, someone might comment that the GWT will generate bloated codes with poor performance and implicit information. However, we utilize the GWT as our platform basis to fast develop out. All of the toolkits, even programming languages, have their own disadvantages, somehow, it is compulsory for us to make use of the GWT to construct our system and avoid potential known bugs.

3.4 Dictionary Binding

It is further confident for the Internet users to bind our scientific method with some open-sourced dictionary gadgets. However, we did not originate such a chore and envisage that we will embrace such a functionality inside our methodology. Furthermore, some of the popular dictionary-oriented functions will be supplemented with no harm.

4 CONCLUSIONS

The further intricate web resources exist, the additionally professional web browsing technologies should be initiated. This paper illustrates a concept for how to extract a web page semantic contentedness and automatically follow the cursor position to organize the likely-words from a sentence for data cleverness. Such a web browsing concept could be carried out with a few cross-browser techniques. We conceive this conception will be popular with any other diversified form in the future browsers. However, this thought will be another important step for human-computer interaction, particularly, for minimizing the time expense and keeping the likely keywords for the time of web surfing.

ACKNOWLEDGEMENTS

The author would like to acknowledge all of the attending students, Chi-Heng Chung, Jun-Shen Chen, Wen-Chen Jian, et al., for their collaborative suggestions in the Software Engineering Department, National Kaohsiung Normal University from September 2008 to January 2009.

REFERENCES

Ding, L., Finin, T., Joshi, A., Peng, Y., Pan, R., and Reddivari, P. (2005). Search on the semantic web. In *Computer, vol. 38, no. 10, pp. 62-69*. IEEE.

Goth, G. (2007). Will the semantic web quietly revolutionize software engineering. In *IEEE Software, vol. 27, no. 4, pp. 100-105*. IEEE.

Hendler, J. (2008). Web 3.0: Chicken farms on the semantic web. In *Computer, vol. 41, no. 1, pp. 17-19*. IEEE.

InternetWorldStats.com (2008). *The Internet World Stats: Usage and Population Statistics*. Miniwatts Marketing Group, http://www.internetworldstats.com/stats.htm.

Missikoff, M., Navigli, R., and Velardi, P. (2002). Integrated approach to web ontology learning and engineering. In *Computer, vol. 35, no. 11, pp. 60-63*. IEEE.

Nano, O. and Zisman, A. (2007). Realizing service-centric software systems. In *IEEE Software, vol. 27, no. 6, pp. 28-30*. IEEE.

Oren, E., Haller, A., Hauswirth, M., Heitmann, B., Decker, S., and Mesnage, C. (2007). A flexible integration framework for semantic web 2.0 applications. In *IEEE Software, vol. 27, no. 5, pp. 64-71*. IEEE.

Sleater, D. D. and Temperley, D. (1993). Parsing english with a link grammar. In *Proc. of the thrid International Workshop on Parsing Technologies, pp. 1-14.*

Vitvar, T., Zaremba, M., Moran, M., Zaremba, M., and Fensel, D. (2007). Sesa: Emerging technology for service-centric environments. In *IEEE Software, vol. 27, no. 6, pp. 56-67*. IEEE.