A Vietnamese Information Retrieval System for Product-Price

Tien-Thanh Vu and **Dat Quoc Nguyen**
Faculty of Information Technology
University of Engineering and Technology
Vietnam National University, Hanoi
{tienthanh_dhcn, datnq}@vnu.edu.vn

***Abstract*—A price information retrieval (IR) system allows users to search and view differences among prices of specific products. Building product-price driven IR system is a challenging and active research area. Approaches entirely depending products information provided by shops via interface environment encounter limitations of database. While automatic systems specifically require product names and commercial websites for their input. For both paradigms, approaches of building product-price IR system for Vietnamese are still very limited. In this paper, we introduce an automatic Vietnamese IR system for product-price by identifying and storing Xpath patterns to extract prices of products from commercial websites. Experiments of our system show promising results.**

***Keywords*-Data mining; Vietnamese Information Retrieval System; Product Information Extraction;**

I. Introduction

A price information retrieval (IR) system allows users to search and view differences between prices of specific products. The system mainly focuses on collecting and updating price information of products crawled from commercial websites. There are generally two main approaches to build a product-price IR system:

- The first bases on interaction between shops and the product-price IR system, in which the system creates an interface environment allowing shops to directly provide product-price information to system. This system type encounters limitations of database in entire dependence on the shops. Because the price always changes over time, it requires price information to be constantly updated to the database.
- The other automatically updates the IR system's database by crawling on commercial websites of shops to extract product-price information. However, this system type requires that product names must be firstly provided and commercial websites must be specified.

In this paper, we introduce a price-driven Vietnamese IR system for products in handling above mentioned drawbacks. With a small number of initial seed product names, our system's front-end component automatically identifies related commercial websites and corresponding Xpath patterns. Then the back-end component uses the related websites and Xpath patterns to collect and update the database of names and prices from crawled products.

The rest of paper is organized as follows: in section II, we provide some related works. We describe our system and our experiments in section III and section IV respectively. The conclusion and future works will be presented in section V.

II. Related works

There have already existed numerous shopping search engines, but they mostly require product-information to be collected and updated manually. PriceScan[1] and GoogleProduct[2] show products from a manually updated database. Kelkoo[3] and Yahoo! Shopping[4] utilize database frameworks where merchants submit their products to be manually classified according to a defined structure. Recently, some Vietnamese shopping search engines have been presented such as: www.vatgia.com, www.aha.vn. But all of them is built according to the first main approach shown in the introduction.

[1] www.http://www.pricescan.com
[2] http://www.google.com/prdhp
[3] http://www.kelkoo.co.uk
[4] http://shopping.yahoo.com

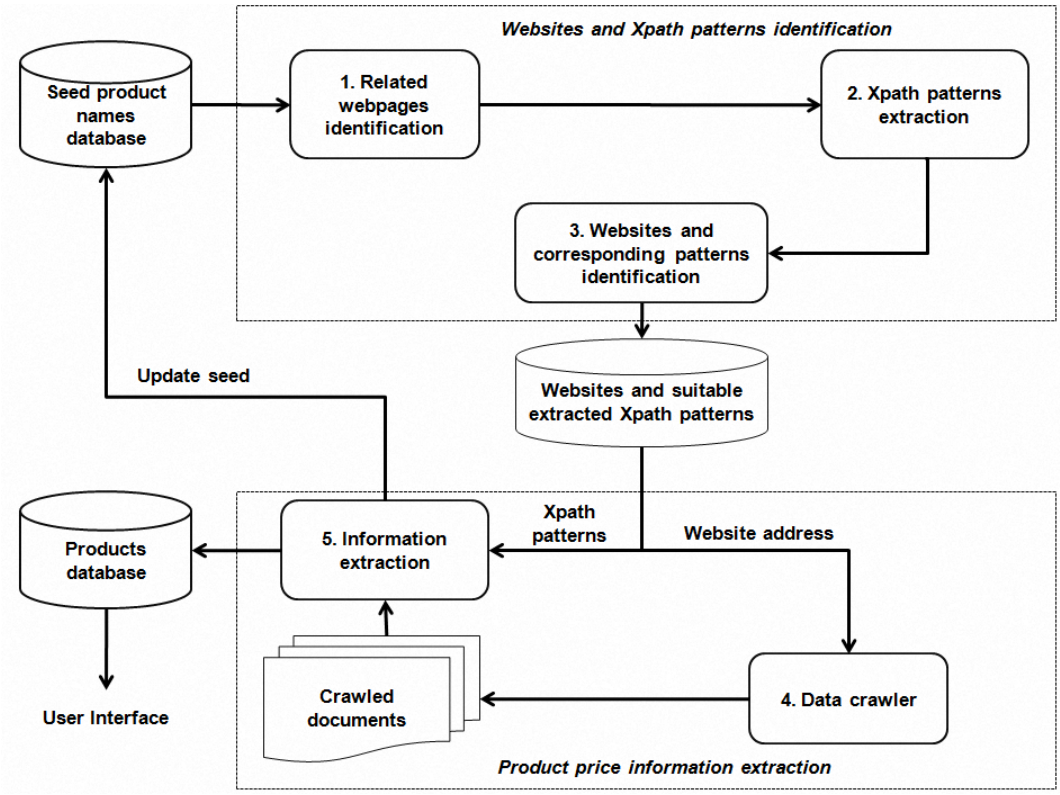

Figure 1. Architecture of our price IR system.

The related works to our approach come from primary field of information extraction from semi-structured webpages. Kushmerick et al [1], Muslea et al.[2], Freitag and Kushmerick [3], Cohen et al.[4] introduced and improved wrapper induction method which generates extraction rules in using machine learning approach. From a few training webpages which manually predetermine the target-items, the method learns to extract rules. The rules then are applied to detect target-items from other pages.

Nguyen et al. [5] proposed an approach to automatically extract primary text content of webpages by identifying and storing templates representing the Xpath structure of text content blocks in websites. Carlson and Schafer [6] described bootstrapping information extraction method which only annotates 2–5 webpages over 4–6 websites. The obtained results significantly outperform the baseline approach with the extraction accuracy of 83.8% on job offer websites and 91.1% on vacation rental websites. Crescenzi et al. [7] presented Roadrunner system which automatically extract information by comparing structure of web pages in requirement of extracted data to be labelled by user.

Zhang et al.[8] described an ontology-based e-commerce product information retrieval framework and presented an ontology-based adaptation of the classical Vector Space Model in considering the weight of product's attributes.

III. Our Vietnamese IR system for product-price

In this section, we describe our product-price information retrieval system . Figure 1 shows our price IR system's architecture. Our system contains two components front-end and back-end. The front-end takes input of seed product names to automatically identify suitable websites and Xpath patterns. The back-end component of product-price information extraction crawls data from URLs in the suitable websites and uses Xpath patterns to extract names and prices information of products in crawled data. The extracted information will be updated into databases of products and seed product names.

A. The front-end component of websites and Xpath patterns identification

The font-end component consists of three modules of "related webpages identification", "Xpath patterns extraction", and "websites and corresponding patterns identification".

1) Related webpages identification module: This module takes a set of seed product names as the input and returns webpages relating to the product names.

Based on specific characteristics of commercial websites, we create particular queries matching product names to Google search engine by utilizing some defined templates. For example: instead of using query *"ipad 2"*, the query *"ipad 2"* + *"vnđ or usd"* is automatically generated in the use of template ***"product_name"*** + *"vnđ or usd"*, and it is sent to Google search engine. All top five webpages of returned results by the Google are from commercial domains.

2) Xpath patterns extraction module: The input of this module is a product name and a related webpage returned by Google search engine. The output is actual price and Xpath patterns to be used to detect product names and the actual prices.

For example, with given product name of *"Nokia 1200"* and one of related webpages identified from the previous module, the patterns extraction module returns results of *"VNĐ 540.000"* (figure 2) and Xpath patterns of *"HTML → BODY → TABLE[1] → TR[1] → TD[1] →* ***product_name****"* and *"HTML → BODY → TABLE[1] → TR[2] → TD[2] →* ***actual_product_price****"*.

Because webpages on the same website usually have similar structures, we can use these Xpath patterns to extract product names and corresponding actual prices from other webpages.

Figure 2. An example of actual price extraction.

The Xpath patterns extraction module has 2 sub-modules:

- **The first sub-module** identifies leaf node in Document Object Model (DOM) tree corresponding with HTML source code of the input related webpage, in which the node contains the text string matching the input product name. The first sub-module generates Xpath pattern by using traversal path from root node of DOM tree to the detected leaf node.
- **The second sub-module** firstly find the leaf node in the DOM tree in which the node contains text string of actual price, and then the second generates corresponding Xpath pattern. The module detects the node containing text string catching "actual price" through following steps:

- *Step 1*: Detect all text strings representing numbers in the input webpage by employing basic regular expressions. For example, in figure 2, extracted text strings are *"1200"*, *"590.000"*, *"540.000"* and *"100.000"*.
- *Step 2*: From extracted text strings via step 1, the module identifies all text strings describing maybe-actual prices through prefix, suffix, and excluding rules:

  Prefix rule: A number represents a product-price if it is preceded by "Giá$_{price}$" or "VNĐ$_{Vietnam\ dong}$",...

  Suffix rule: A number represents a product-price if is followed by "VNĐ$_{Vietnam\ dong}$", "USD", "Đ$_{dong}$", "$",...

  Excluding rules: A text string does not represent an actual price if it is preceded by "Giá cũ$_{Old\ price}$", "Giá thị trường$_{Market\ price}$", "Tiết kiệm$_{Save}$",... A text string does not represent an actual price if it is stored by DOM tree nodes of tags <strike> or <s>. For example, in figure 2, text string *"VNĐ 590.000"* is not actual price because the text string belongs to tree node of tag <strike>. Text string *"VNĐ 100.000"* followed by "Tiết kiệm$_{Save}$" is not actual price.
- *Step 3*: Determine the actual price if there are some maybe-actual prices. It needs to examine relationship between name and actual price of product. The relationship means that product's name and product's actual price are held by two closet nodes of DOM tree. It is a specific characteristic of commercial webpages.

For example: with the Xpath pattern *HTML* $\rightarrow$ *BODY* $\rightarrow$ *TABLE* $\rightarrow$ *TR* $\rightarrow$ *TD* $\rightarrow$ *DIV[1]* $\rightarrow$ ***product_name*** generated from the first sub-module to extract the input product name, and a Xpath pattern corresponding with a maybe-actual price *HTML* $\rightarrow$ *BODY* $\rightarrow$ *TABLE* $\rightarrow$ *TR* $\rightarrow$ *TD* $\rightarrow$ *DIV[2]* $\rightarrow$ *FONT* $\rightarrow$ ***product_price***. The similar-measure is 5 overlap steps *HTML[1]* $\rightarrow$ *BODY[2]* $\rightarrow$ *TABLE [3]*$\rightarrow$ *TR[4]* $\rightarrow$ *TD[5]*. The Xpath pattern to extract price, that has highest similar-measure in comparison with the Xpath pattern used to extract input product name, is selected as output pattern to extract actual price.

3) Websites and corresponding patterns identification module: This module returns commercial websites and suitable Xpath patterns to be used to generate names and actual prices of products from the themselves. The module counts number webpages from each website in which the webpages have same identified Xpath patterns determined the previous module. If the number is greater than a given threshold, the website is considered as a commercial website and the corresponding Xpath patterns are suitable patterns.

B. The back-end component of product-price information extraction

In this component, we focus on two modules Data crawler and Information extraction. The component takes front-end's output as input of identified commercial websites and suitable Xpath patterns matching with each website. HTML documents from the websites will be collected in the use of Data crawler module via browsing hyper-links in each crawled document.

The information extraction module uses the input of collected HTML documents and suitable Xpath patterns to extract information of product names and actual product-prices. Extracted information then will be updated into Products database and Seed product names database (figure 1).

IV. EXPERIMENTS

We built our system on computer of Intel Celeron@CPU 2.66GHz and RAM 768MB. With initial set of 334 seed product names from many product types such as mobile phone, computer, camera, jewellery, household items,... in 30 hours, our system collected 47856 products from 125 determined commercial websites in which 34012 products are unique. For example, "Lenovo ThinkPad T61" and "IBM T61" are considered as the same one while "Nokia 1200 black" and "Nokia 1200 white" are different. In order to clearly evaluate our system's modules, we present some experiments as follows.

A. Experiment of "Related webpages identification"

To evaluate the template *"**product_name**" + "VNĐ or USD"* that we employed to create queries, we randomly selected products of "Nokia 1200", "Lenovo Thinkpad t61" and "Canon PowerShot G10". Table I shows the number of commercial webpages containing product name and its actual price, in top 10, 30 and 100 returned webpages by using Google Search Engine. Other returned results by Google belong to webpages of news, forums,...

Table I
NUMBER OF COMMERCIAL WEBPAGES RETURNED BY GOOGLE SEARCH ENGINE

Product name	Number of related webpages by Google	Number of commercial webpages
Nokia 1200	10	8
	30	23
	100	68
Lenovo Thinkpad t61	10	10
	30	23
	100	43
Canon PowerShot G10	10	9
	30	19
	100	45

B. Experiment of actual price extraction in "Xpath patterns extraction" module

To right examine extraction-ability of this module, we used the commercial webpages determined in the previous experiment (table I). In this experiment, we consider $F_{measure}$ as a metric to evaluate the accuracy of price extraction as presented in table II.

$$F_{measure} = \frac{2 * Recall * Precision}{Recall + Precision}$$

Precision is defined as the ratio between the number of extracted actual-prices and the total number of detected prices, while *Recall* is defined as the ratio between the number of extracted actual-prices and the actual number of actual-prices.

Table IV
ACCURACY OF PRODUCT'S N NAME AND PRICE EXTRACTION

Website	Number of crawled webpages	Number of commercial webpages	Number of pairs of extracted product name and corresponding actual price
www.dienthoaididong.com.vn	850	792	743 (93.81 %)
www.trananh.vn	800	711	416 (58.5 %)

Table II
THE ACCURACY OF PRICE EXTRACTION

Product name	Recall	Precision	F-measure
Nokia 1200	8/8 (1.0)	8/8 (1.0)	100 %
	23/23 (1.0)	23/26 (0.88)	93.88 %
	67/68 (0.99)	67/70 (0.96)	97.10 %
Lenovo Thinkpad t61	9/10 (0.9)	9/10 (0.9)	90 %
	22/23 (0.96)	22/25 (0.88)	91.67 %
	40/43 (0.93)	40/46 (0.87)	89.89 %
Canon PowerShot G10	9/9 (1.0)	9/9 (1.0)	100 %
	18/19 (0.95)	18/21 (0.86)	90 %
	44/45 (0.98)	44/50 (0.88)	92.63 %

Table III
ACCURACY OF COMMERCIAL WEBSITES IDENTIFICATION

Top results of Google	Identified websites	Accuracy
10	www.123mua.com.vn www.vatgia.com www.vinacms.vn www.chodientu.vn	100 %
100	www.123mua.com.vn www.vatgia.com www.vinacms.vn www.chodientu.vn www.enbac.com www.quangcaosanpham.com www.aha.vn www.dienthoaididong.com.vn www.trananh.vn	100 %

C. Experiment of "commercial websites identification"

For initial set of 4 products of "Nokia 1200", "Nokia e71 white steel", "Nokia 1202" and "Nokia 6300 silver" and a defined threshold of 3 to determine commercial websites, table III gives accuracy of 100% for the first component on both cases of taking top 10 and 100 related webpages returned by Google in the first module of our system.

D. Experiment of "information extraction" module

This experiment shows our evaluation in the use of identified Xpath patterns to extract names and prices of products. From the output of the front-end component in taking the set of 4 products as input that is described in the "commercial websites identification" experiment, we selected two websites *www.dienthoaididong.com.vn* and *www.trananh.vn* and their corresponding suitable Xpath patterns to perform the evaluation.

We randomly crawled a number of webpages per each selected website by "Data crawler" module, in which there are many webpages coming from website's news and forum. We only calculated the accuracy based on number of commercial webpages. Table IV presents promising results that the information extraction module well performed on the website www.dienthoaididong.com.vn. The website www.trananh.vn has different Xpath structures for representing different product categories such as computer, camera, household items,... in HTML documents, therefore, with 4 given seed product names only belonging to the category of mobile phones, 416 extracted products from www.trananh.vn only belong to the mobile phone category. Consequently, the returned result is not high. It is easy to improve the result by taking seed products from all kinds of categories.

V. CONCLUSION

We believe on fast scalability of our system. Our system can identify more sites and Xpath patterns depending on the number of initial seed product names. Because extracted product names returned by information extraction module always are updated into the seed products database, the database always is expanded. In addition, it is possible for our proposed system's architecture to adapt to a new language by changing the rules according to the new one.

In this paper, we introduce an automatic product-price information retrieval system for Vietnamese commercial sites. With a small number of seed product names, our system automatically detects commercial sites, generates corresponding Xpath patterns. Our

system then uses identified information to extract name and actual price of crawled products.

The experiment results are promising; with 334 initial product names, our system determined 125 commercial sites and collected 47.856 products in 30 hours. In the future, we will extend our system's rules driving to collect information of size, weight, guarantee period, and other features of products.

ACKNOWLEDGEMENT

The authors would like to acknowledge Vietnam National Foundation for Science and Technology Development (NAFOSTED) for their financial support to present the work at the conference.

REFERENCES

[1] N. Kushmerick, D. Weld, and R. Doorenbos, "Wrapper induction for information extraction," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 1997.*, 1997.

[2] I. Muslea, S. Minton, and C. Knoblock, "A hierarchical approach to wrapper induction," in *Proceedings of the third annual conference on Autonomous Agents*, 1999, pp. 190–197.

[3] D. Freitag and N. Kushmerick, "Boosted wrapper induction," in *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, 2000, pp. 577–583.

[4] W. W. Cohen, M. Hurst, and L. S. Jensen, "A flexible learning system for wrapping tables and lists in html documents," in *Proceedings of the 11th international conference on World Wide Web*, 2002, pp. 232–241.

[5] D. Q. Nguyen, D. Q. Nguyen, S. B. Pham, and T. D. Bui, "A fast template-based approach to automatically identify primary text content of a web page," in *Proceedings of the 2009 International Conference on Knowledge and Systems Engineering*, ser. KSE '09, 2009, pp. 232–236.

[6] A. Carlson and C. Schafer, "Bootstrapping information extraction from semi-structured web pages," in *Proceedings of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases - Part I*, 2008, pp. 195–210.

[7] V. Crescenzi, G. Mecca, and P. Merialdo, "Roadrunner: Towards automatic data extraction from large web sites," in *Proceedings of 27th International Conference on Very Large Data Bases*, 2001, pp. 109–118.

[8] L. Zhang, M. Zhu, and W. Huang, "A framework for an ontology-based e-commerce product information retrieval system," *JCP*, vol. 4, no. 6, pp. 436–443, 2009.