Notice of Violation of IEEE Publication Principles

"Architecture and Implementation of an Object-level Vertical Search"
by Jianfeng Zheng; Zaiqing Nie
in the Proceedings of the International Conference on New Trends in Information and Service Science, (NISS), June 2009, pp. 264-268

After careful and considered review of the content and authorship of this paper by a duly constituted expert committee, this paper has been found to be in violation of IEEE's Publication Principles.

This paper is a verbatim copy of the paper cited below. The lead author, Jianfeng Zheng, submitted the copied paper without the knowledge or permission of the coauthor, Zaiqing Nie.

Due to the nature of this violation, reasonable effort should be made to remove all past references to this paper, and future references should be made to the following article:

"Object-level Vertical Search"
by Zaiqing Nie, Ji-Rong Wen, and Wei-Ying Ma
in the Proceedings of the Third Biennial Conference on Innovative Data Systems Research (CIDR), January 2007

2009 International Conference on New Trends in Information and Service Science

Architecture and Implementation of an Object-level Vertical Search

Jianfeng Zheng
School of Economics and Management
BUPT
Beijing,China
kezheng@microsoft.com

Zaiqing Nie
Microsoft Research Asia
Microsoft
Beijing, China
znie@microsoft.com

Abstract

Current mobile web search engines essentially conduct document-level ranking and retrieval. However, structured information about real-world objects embedded in static WebPages and online databases exists in huge amounts. We explore a new paradigm to enable web search at the object level in this paper, extracting and integrating web information for objects relevant to a specific application domain. We then rank these objects in terms of their relevance and popularity in answering user queries. We introduce the overview and core technologies of object-level vertical search engines that have been implemented in one working systems: Windows Live Product Search on mobile (http://m.live.com).

Keywords—Web Information Extraction, Information Integration, Web Search, Object-level Ranking

1. Introduction

Table 1. Object-Level Search vs. Page-Level Search

	Page-Level Search	Object-Level Search
Technology	Information Retrieval; Pages as Retrieval Units	Database; Machine Learning; Objects as Retrieval Units
Pros	Ease of Authoring; Ease of Use	Powerful Query Capability; Direct Answer; Aggregate Answer
Cons	Limited Query Capability	Where and How to Get the Objects?

In Table 1, we compare object-level and page-level search. In page-level search, WebPages are the basic retrieval units, and the information in a page is treated as a bag of words. Information retrieval technologies are used as core technologies to answer user queries, while object-level search uncovers structured information about real-world objects that are the retrieval units. One obvious advantage of object-level search is its capability of answering complex queries with direct and aggregate answers because of the availability of semantics defined by the object schema. Otherwise, it could take one several hours to sift through hundreds of WebPages returned by a page-level search engine.

The challenge here is where and how to obtain high-quality structured data needed by an object-level search engine, and how to rank resulting objects to return the most relevant ones.

Until now, the Beta release of Window Live Product Search has incorporated our product page classification and extraction techniques. After the first month running, we have already indexed more than 100,000 sellers, 31,627,416 commercial pages, and 800 million automatically extracted product

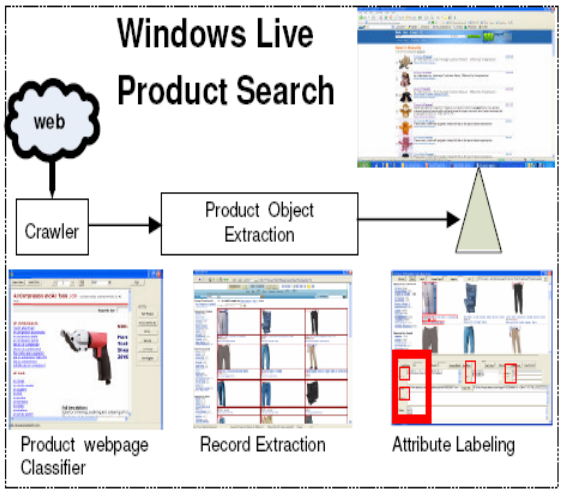

Figure 1. Window Live Product Search with our Product Page Classification and Extraction techniques.

records. We believe we could make Window Live

978-0-7695-3687-3/09 $25.00 © 2009 IEEE
DOI 10.1109/NISS.2009.22

Product Search the largest product catalog in the world by using object-level vertical search technologies.

2. REQUIREMENTS

The requirements for a large-scale object-level vertical search engine are as follows:

- Reliability: Retrieve high quality structured data to generate direct and aggregate answers.
- Completeness: Make sure the data to be as complete as possible to provide trustworthy answers.
- Ranking Accuracy: Optimize ranking mechanism for locating relevant object information.
- Scalability: Get all the information within a vertical domain on the web and store them in local databases.
- In the following sections, we will introduce the system architecture and infrastructure design with these requirements in mind.

3. SYSTEM ARCHITECTURE & CORE TECHNIQUES

Figure 2 shows the brief architecture of an object-level vertical search engine. First, a crawler fetches web data related to the targeted objects within a specific vertical domain, and the crawled data is classified into different categories, such as papers, authors, products, and locations. For each category, a specific entity extractor is built to extract objects from the web data. At the same time, information about the same object is aggregated from multiple different data sources. Once objects are extracted and aggregated, they are put into the object warehouses, and vertical search engines can be constructed based on the object warehouses. Moreover, advanced object-level ranking and mining techniques can be applied to make search more accurate and intelligent.

3.1 Crawler and Classifier

The tasks of the crawler and classifier are to automatically collect all relevant WebPages or documents that contain object information for a specific vertical domain. The crawled WebPages or documents will be passed to the corresponding object extractor for extracting the structured object information and building the object warehouse.

We build a "focused" crawler that uses the page classifier and the existing partial object relationship graph to guide the crawling process. Basically, in addition to the web graph which is used by most page-level crawlers, we employ an object relationship graph to guide our crawling algorithm.

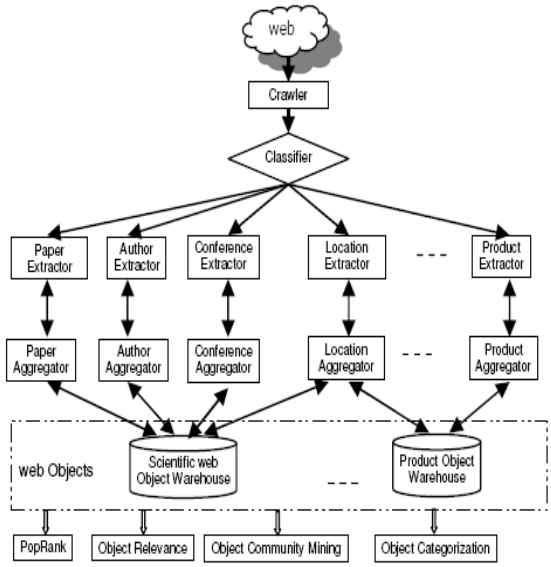

Figure 2. System Architecture

Since the classifier is coupled with the crawler, it needs to be very fast to ensure efficient crawling. Based on our experience in building a classifier for Windows Live Product Search, we found that we could always use some strong heuristics to quickly prune most of irrelevant pages. For example, in our product pages classifier, we can use the price identifiers (such as dollar signs $) to efficiently prune most non-product pages. The average time of our product classifier is around 0.1 millisecond, and its precision is around 0.8, with recall around 0.9.

3.2 Object Extractor

Information (e.g. attributes) about a web object is usually distributed in many web sources and within small segments of WebPages. The task of an object extractor is to extract metadata about a given type of objects from every web page containing this type of objects. For example, for each crawled product web page, we extract name,

image, price and description of each product. If all of these product pages or just half of them are correctly extracted, we will have a huge collection of metadata about real world products that could be used for further knowledge discovery and query answering. Our statistical study on 51,000 randomly crawled WebPages shows that about 12.6 percent are product pages. That is, there are about 1 billion product pages within a search index containing 9 billion crawled WebPages.

However, how to extract product information from WebPages generated by many (maybe tens of thousands of) different templates is non-trivial. One possible solution is that we first distinguish WebPages generated by different templates, and then build an extractor for each template. We say that this type of solution is template-dependent.

3.2.1 Template-Independent Web Object Extraction

We propose template-independent metadata extraction techniques for the same type of objects. Specifically in [7][10][11], we extended the linear-chain Conditional Random Fields (CRFs) [5] which are the state of the art approaches in information extraction taking advantage of the sequencing characteristics to do better labeling.

3.2.2 2D Conditional Random Fields

In order to use the existing linear-chain CRFs for Web object extraction, we have to first convert a two-dimensional object block (i.e. an object block whose elements are two-dimensionally laid out) into a sequence of object elements.

Given the two-dimensional nature of object blocks, how to sequentialize them in a meaningful way could be very challenging. Moreover, as shown by our empirical evaluation, using the two-dimensional neighborhood dependencies (i.e. interactions between labels of an element and its neighbors in both vertical and horizontal directions) in Web object extraction could significantly improve the extraction accuracy.

To better incorporate the two-dimensional neighborhood dependencies, a two-dimensional Conditional Random Field (2D CRF) model is proposed in [10]. We present the graphical representation of the 2D CRF model as a 2D grid (See Figure 3)

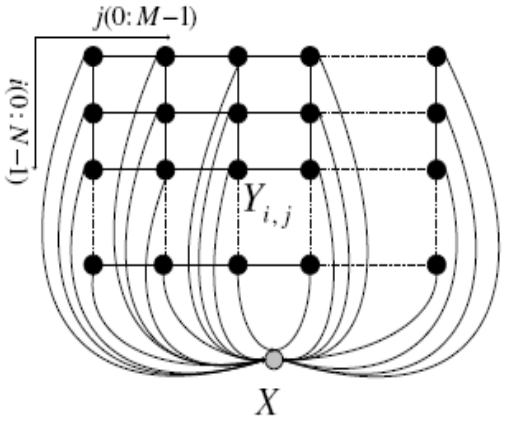

Figure 3. The Graphical Structure of 2D CRFs

And reformulate the conditional distribution by defining some matrix random variables. Then we deduce the forward-backward vectors based on the reformulated conditional distribution for efficient parameter estimation and labeling. Since the sizes of the elements in an object block can be arbitrary, we introduce the concept of virtual states to model an object block as a 2D grid. We compare our model with linear-chain CRF models for product information extraction and the experimental results show that our model significantly outperforms linear-chain CRF models in scenarios with two-dimensional neighborhood dependencies.

3.3 Object Aggregator

Each extracted web object need to be mapped to a real world object and stored into a web data warehouse. To do so, the object aggregator needs to integrate information about the same object and disambiguate different objects.

Some recent work [2] proposes the exploitation of connection via object relationships in the object-relationship graph, in addition to the available object attribute values, for name disambiguation. The assumption behind their approaches is that, if two identical names in different contexts refer to the same object, they are more likely to be strongly connected on the entity relationship graph. Based on this assumption, two identical names are detected as referring to the same object only when the connection strength between them is stronger than a predefined threshold.

3.3.1 Object Identification using Web Connections

We measure the web connection between two object appearances based on the co-occurrences of their contexts in a website (or webpage). The co-occurrence information could be easily obtained by sending the context information as queries to a search engine (e.g., Google, MSN Search). However it is non-trivial to measure the web connection strength based on the co-occurrence information, since co-occurrences in different websites usually indicate quite different connection strengths. To handle these issues, our measure of web connection not only discriminates the relative importance of different websites, but also considers the URL distance between WebPages inside a website.

Our name disambiguation approach considers two object appearances with the same name as the same object once their context-object information is found in a small hub. For the appearance pairs where there is no co occurrence in any small hub, we need to compute the Web connection strength for all their co-occurrences in big hubs. As shown in the above observation, the number of big hubs is relatively limited, so it becomes feasible to train an adaptive connection function which gives suitable weights to the co occurrences in these big hubs.

3.3.2 Object-level Ranking

On the traditional web graph, different pages have different popularity according to their in-links. Technologies such as Page Rank [12] and HITS [3] have been successfully applied to distinguish the popularity of different WebPages through analyzing the link structure in the web graph.

We propose Pop Rank, a method to measure the popularity of web objects in an object graph. It extends the Page Rank model by using different propagation factors for links of different types of relationships. We propose a learning based approach to automatically learn the popularity propagation factors for different types of links using the partial ranking of the objects given by domain experts. The simulated annealing algorithm is used to explore the search space of all possible combinations of propagation factors and to iteratively reduce the difference between the partial ranking from the domain experts and that from our learned model.

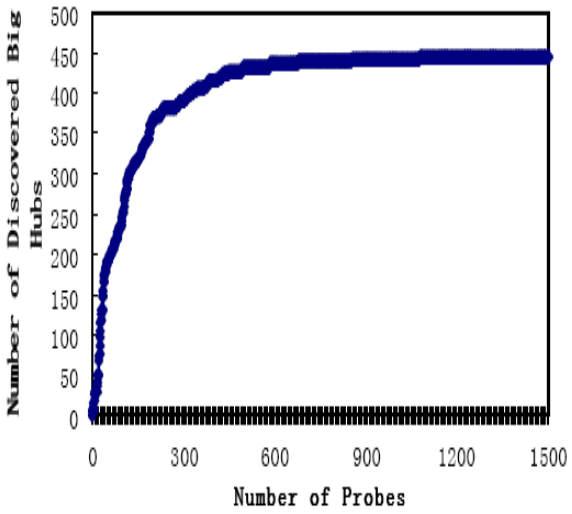

Figure 4. The Growth of Big Hub (>1%) Numbers by Randomly Probing

One major challenging problem facing our learning approach is that it is prohibitively expensive to try hundreds of combinations of feasible factors which are normally needed for us to get a reasonable assignment of the propagation factors. It may take hours to compute the Pop Rank of the objects to test the optimality of a PPF factor assignment. In order to make the learning time manageable, we propose the use of a sub graph of the entire object link graph in the learning process. As soon as we have most of the related objects and their links surrounding the training objects, we should be able to calculate a close approximation of the Pop Rank of these training objects.

4. Conclusion

In this paper, we propose a new paradigm called object-level vertical search to enable web search at the object level.

Specifically, we introduce the system architecture of such an object-level search engine and its core techniques. More importantly, we share our experience in building the real vertical search engines: Window Live Product Search. We are currently working on evaluating the model in a more general way and in other application domains. We believe that our approach is generally applicable for most vertical search domains, such as Yellow Page Search, Blog Search, People Search, Job Search, and Restaurant Search.

5. References

[1] Bekkerman, R., and McCallum, A. Disambiguating Web Appearances of People in a Social Network. In Proc. of the WWW, 2005.

[2] Chen, Z., Kalashnikov, D.V., and Mehrotra, S. Exploiting relationships for object consolidation. In ACM IQIS, 2006.

[3] Kleinberg, J. Authoritative Sources in a Hyperlinked Environment, in Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, 1998

[4] Kushmerick, N. Wrapper induction: efficiency and expressiveness. Artificial Intelligence, 2000,118:15-68.

[5] Lafferty, J., McCallum, A., and Pereira, F. Conditional random fields: Probabilistic models for segmenting and labelling sequence data. In Proc. of ICML, 2001.

[6] Nie, Z., Zhang, Y., Wen, J.-R., and Ma, W.-Y. Object-level Ranking: Bringing Order to web Objects. In Proc. WWW, 2007.

[7] Nie, Z., Wu, F., Wen, J.-R., and Ma, W.-Y. Extracting Objects from the Web. In Proc. of ICDE. 2006.

[8] On, B., Elmacioglu, E., Lee, D., Kang, J., and Pei, J. An Effective Approach to Entity Resolution Problem Using QuasiClique and its Application to Digital Libraries. In JCDL 2006.

[9] Page, L., Brin, S., Motwani, R.,Winograd, T. The PageRank Citation Ranking: Bringing Order to the Web. Technical Report, Stanford Digital Library Technologies Project, 1998

[10] Zhu, J., Nie, Z., Wen, J.-R., Zhang, B., and Ma, W.-Y. 2D Conditional Random Fields for web Information Extraction. In Proc. of ICML, 2007.

[11] Skounakis, M., Craven, M., and Ray S. Hierarchical Hidden Markov Models for Information Extraction. In Proc. Of IJCAI, 2003.

[12] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. Modern Information Retrieval. Addison-Wesley Publishers, 1999.

[13] Deng Cai, Xiaofei He, Ji-Rong Wen, and Wei-Ying Ma. Block-Level Link Analysis. In Proceedings of SIGIR, 2004.

[14] J.P. Callan. Distributed information retrieval. In Advances in Information Retrieval: Recent Research from the Center for Intelligent Information Retrieval, edited by W. Bruce Croft. Kluwer Academic Publisher, pp. 127-150, 2000.

[15] Abdur Chowdhury, Mohammed Aljlayl, Eric Jensen, Steve Beitzel, David Grossman and Ophir Frieder. Linear Combinations Based on Document Structure and Varied Stemming for Arabic Retrieval. In The Eleventh Text REtrieval Conference (TREC 2002), 2003.

[16] Charles L.A. Clarke. Controlling Overlap in Content-Oriented XML Retrieval. In Proceedings of the SIGIR,2005.

[17] Nick Craswell, David Hawking and Trystan Upstill. TREC12 Web and Interactive Tracks at CSIRO. In The Twelfth Text Retrieval Conference(TREC 2003), 2004.

[18] Ronald Fagin, Ravi Kumar, Kevin S. McCurley, Jasmine Novak, D. Sivakumar, John A. Tomlin and David P.Williamson. Searching the Workplace Web. In Proceedings of the Twelfth International World Wide Web Conference, 2003.

[19] Hui Fang, Tao Tao and ChengXiang Zhai. A Formal Study of Information Retrieval Heuristics. In Proceedings of SIGIR, 2004. [12] Norbert Fuhr. Probabilistic Models in Information Retrieval. The computer Journal, Vol.35, No.3, pp. 243-255.

[20] David Hull. Using Statistical Testing in the Evaluation of Retrieval Experiments. In Proceedings of the ACM SIGIR, 1993.

[21] Zaiqing Nie, Yuanzhi Zhang, Ji-Rong Wen and Wei-Ying Ma. Object-Level Ranking: Bringing Order to Web Objects. In Proceedings of the 14th international World Wide Web Conference (WWW), 2005.

[22] Zaiqing Nie, Ji-Rong Wen and Wei-Ying Ma. Object-Level Vertical Search. To appear by the Third Biennial Conference on Innovative Data Systems Research (CIDR),2007.

[23] Yiming Yang and Xin Liu. A re-examination of text categorization methods. In Proceedings of the ACM SIGIR, 1999.

[24] Jun Zhu, Zaiqing Nie, Ji-Rong Wen, Bo Zhang, Wei-Ying Ma. 2D Conditional Random Fields for Web Information Extraction. In Proceedings of the 22nd International Conference on Machine Learning (ICML), 2005.

[25] Jun Zhu, Zaiqing Nie, Ji-Rong Wen, Bo Zhang, Wei-Ying Ma. Simultaneous Record Detection and Attribute Labeling in Web Data Extraction. ACM Discovery and Data Mining (KDD), 2007.

[26] Zuobing Xu, Ram Akella, Active Relevance Feedback for Difficult Queries. To be published in Proceedings of ACM 17th Conference on Information and Knowledge Management (CIKM) 2008

[27] Zuobing Xu, Ram Akella, Bayesian Logistic Regression Model for Active Relevance Feedback. In Proceedings of the 31st ACM SIGIR Conference, 2008.

[28] Zuobing Xu, Ram Akella, New Probabilistic Retrieval Model Based on the Dirichlet Compound Multinomial Distribution In Proceedings of the 31st SIGIR Conference, 2008.