Large-Scale Copy Detection

Xin Luna Dong
AT&T Labs–Research
lunadong@research.att.com

Divesh Srivastava
AT&T Labs–Research
divesh@research.att.com

ABSTRACT

The Web has enabled the availability of a vast amount of useful information in recent years. However, the web technologies that have enabled sources to share their information have also made it easy for sources to copy from each other and often publish without proper attribution. Understanding the copying relationships between sources has many benefits, including helping data providers protect their own rights, improving various aspects of data integration, and facilitating in-depth analysis of information flow.

The importance of copy detection has led to a substantial amount of research in many disciplines of Computer Science, based on the type of information considered, such as text, images, videos, software code, and structured data. This tutorial explores the similarities and differences between the techniques proposed for copy detection across the different types of information. We also examine the computational challenges associated with large-scale copy detection, indicating how they could be detected efficiently, and identify a range of open problems for the community.

Categories and Subject Descriptors

H.3.5 [**Online Information Services**]: Data sharing

General Terms

Algorithms

Keywords

Copy detection, data integration

1. INTRODUCTION

The Web has enabled the availability of a vast amount of useful information in recent years. In a wide variety of domains, ranging from science and technology to arts and entertainment, from business and government to leisure and travel, there are a huge number of web sources that seek to provide information, in the form of web pages, images, video, structured data, software, etc., to a wide spectrum of users. However, the web technologies that have enabled sources to share their information have also made it easy for sources to copy from each other and often publish without proper attribution.

The copying relationships can be complex: some sources act as hubs and aggregate information from multiple sources; some provide only a small amount of information independently, copying the rest from other sources; some sources are well known and widely copied by many other sources. Understanding the copying relationships between sources has many benefits. First, information is valuable and many sources have put in a lot of money and effort in collecting and cleaning their information, so they may want to understand such relationships for business purposes and also to protect their own rights. Second, considering copying relationships can help web services such as search engines, question answering systems and data integration systems provide results of higher quality. Finally, identifying provenance of information can be critical for an in-depth analysis of information flowing among virtual communities on the web, including social networking sites, blogs, video-sharing sites, etc.

The importance of copy detection has led to a substantial amount of research in many disciplines of Computer Science, based on the type of information considered. The Information Retrieval community has devoted considerable effort to finding plagiarism, near-duplicate web pages and text reuse (see, e.g., [24, 16, 32, 13, 2, 25]). The Multimedia community has considered techniques for copy detection of images and video, especially in the presence of distortion (see, e.g., [14, 15, 23, 22, 19, 21, 18]). The Software Engineering community has examined techniques to detect clones of software code (see, e.g., [7, 20, 1, 17, 11, 30, 29]). Finally, the Database community has focused on mining and making use of overlapping information between structured sources (see, e.g., [27, 28, 5]), finding copies of documents across multiple databases (see, e.g., [6, 12, 31]), and more recently on copying of structured data across sources (see, e.g., [3, 9, 10, 8, 4, 26]).

In this tutorial, we explore the similarities and differences between the techniques proposed for copy detection across the different types of information. We do this with illustrative examples that would be of interest to data management researchers and practitioners. We also examine the computational challenges associated with large-scale copy detection, indicating how they could be detected efficiently, and identify a range of open problems for the community.

SIGMOD'11, June 12–16, 2011, Athens, Greece.
Copyright 2011 ACM 978-1-4503-0661-4/11/06 ...$10.00.

2. TUTORIAL OUTLINE

Our tutorial is example driven, and organized as follows.

2.1 Information Copying Examples

The tutorial will start with a variety of real-world examples illustrating the prevalence of information copying on the web. The examples will highlight the adverse impacts of copying, especially when the information is of questionable accuracy. For example, an obituary of Apple founder Steve Jobs was published and sent to thousands of corporate clients on Aug 28, 2008, before it was retracted.[1] Such false information can often result in considerable damage; for example, the recent incorrect news about United airlines filing for a second bankruptcy sent its shares tumbling, before the error was corrected.[2] The Web also makes it easy to rapidly spread rumors, which take a long time to die down. For example, the rumor from the late 1990s that the MMR vaccine given to children in Britain was harmful and linked to autism caused a significant drop in MMR coverage, leading autism experts to spend years trying to dispel the rumor.[3] Similarly, the upcoming experiments at the Large Hadron Collider (LHC) have sparked fears among the public that the LHC particle collisions might produce dangerous microscopic black holes that may mean the end of the world.[4]

2.2 Common Themes in Copy Detection

Next, we overview the common themes underlying copy detection techniques for various types of data.

The first common theme is to detect unexpected sharing of data fragments under the no-copy assumption. For texts, such unexpected sharing can be on sentences, on characteristic paragraphs, or on writing styles; for images and videos, such unexpected sharing can be on portions of images, or on frames in videos; for code, such unexpected sharing can be on sequences of words or tokens, on the tree-based structure, on the program dependency graph (semantics of the code), or on metrics such as number of lines of code per function and number of function calls; for relational data, such unexpected sharing can be on rarely-provided items, on false or unpopular values, on formatting styles, and so on.

The second common theme is to be tolerant to distortion or modification of copied information. There can be various kinds of modification, including paraphrasing for texts, distortion for images and videos, parameter renaming and function reordering for code, value changes and reformatting for relational data.

2.3 Copy Detection for Unstructured Data

In this unit, we present a variety of techniques proposed for detection of plagiarism in information represented as text (see, e.g., [24, 16, 32, 13, 2, 25]), images and video (see, e.g., [14, 15, 23, 22, 19, 21, 18]). At the heart of these techniques are scalable algorithms for similarity detection, and we identify common techniques explored across the different types of information.

2.4 Copy Detection for Structured Data

In this unit, we present a variety of techniques proposed for copy detection when the information has a richer structure than simple text. We consider approaches for both software code and relational databases.

In particular, for software code, we highlight the use of tree structure and dependency graph for copy detection (see, e.g., [7, 20, 1, 17, 11, 30, 29]). For relational databases, we differentiate between techniques that simply find overlapping information between structured sources (see, e.g., [27, 28, 5]) and those that are able to detect evidence of copying (see, e.g., [3, 9, 10, 8, 4, 26]). We will highlight the role of source quality metrics like accuracy and coverage in copy detection.

2.5 Open Problems

We will present open problems in copy detection. For relational data, such open problems include improving scalability of copying detection, detecting copying in an open world where there can be hidden sources, and combining copy detection with other integration techniques such as schema mapping and record linkage for better detection results. More broadly, it is a challenging problem to perform web-scale copy detection, and exploit evidence from various types of information for detecting copying between structured and unstructured sources.

3. CONCLUSIONS

Copying of information is prevalent on the Web, and understanding the copying relationships between sources is very important. Our tutorial explores the similarities and differences between the techniques proposed for large-scale copy detection across different types of information, such as text, images, videos, software code, and structured data.

We expect two main learning outcomes from this tutorial. In the short term, we expect that this tutorial, by comparing and contrasting the techniques used by different communities for copy detection, will enable the audience to gain a unified understanding of the topic. Taking a more long-term view, we hope that it will foster interactions between researchers across these multiple disciplines to investigate and develop more comprehensive and scalable techniques for copy detection on the web.

4. REFERENCES

[1] S. Bellon, R. Koschke, G. Antoniol, J. Krinke, and E. Merlo. Comparison and evaluation of clone detection tools. *IEEE Trans. Software Eng.*, 33(9):577–591, 2007.

[2] M. Bendersky and W. B. Croft. Finding text reuse on the web. In *WSDM*, pages 262–271. 2009.

[3] L. Berti-Equille, A. D. Sarma, X. Dong, A. Marian, and D. Srivastava. Sailing the information ocean with awareness of currents: Discovery and application of source dependence. In *CIDR*. 2009.

[4] L. Blanco, V. Crescenzi, P. Merialdo, and P. Papotti. Probabilistic models to reconcile complex data from inaccurate data sources. In *CAiSE*, volume 6051 of *Lecture Notes in Computer Science*, pages 83–97. Springer, 2010.

[5] J. Bleiholder, S. Khuller, F. Naumann, L. Raschid, and Y. Wu. Query planning in the presence of overlapping

[1] http://www.telegraph.co.uk/news/newstopics/howaboutthat/2638481/ Steve-Jobs-obituary-published-by-Bloomberg.html

[2] http://gawker.com/5047763/how-robots-destroyed-united-airlines

[3] http://www.guardian.co.uk/society/2008/apr/12/health.children

[4] http://en.wikipedia.org/wiki/Large_Hadron_Collider#Safety_of_particle_collisions

sources. In *EDBT*, volume 3896 of *Lecture Notes in Computer Science*, pages 811–828. Springer, 2006.

[6] S. Brin, J. Davis, and H. Garcia-Molina. Copy detection mechanisms for digital documents. In *SIGMOD Conference*, pages 398–409. ACM Press, 1995.

[7] X. Chen, B. Francia, M. Li, B. McKinnon, and A. Seker. Shared information and program plagiarism detection. *IEEE Transactions on Information Theory*, 50(7):1545–1551, 2004.

[8] X. Dong, L. Berti-Equille, Y. Hu, and D. Srivastava. Global detection of complex copying relationships between sources. *PVLDB*, 3(1):1358–1369, 2010.

[9] X. L. Dong, L. Berti-Equille, and D. Srivastava. Integrating conflicting data: The role of source dependence. *PVLDB*, 2(1):550–561, 2009.

[10] X. L. Dong, L. Berti-Equille, and D. Srivastava. Truth discovery and copying detection in a dynamic world. *PVLDB*, 2(1):562–573, 2009.

[11] R. Falke, P. Frenzel, and R. Koschke. Empirical evaluation of clone detection using syntax suffix trees. *Empirical Software Engineering*, 13(6):601–643, 2008.

[12] H. Garcia-Molina, L. Gravano, and N. Shivakumar. dscam: Finding document copies across multiple databases. In *PDIS*, pages 68–79. IEEE Computer Society, 1996.

[13] O. A. Hamid, B. Behzadi, S. Christoph, and M. R. Henzinger. Detecting the origin of text segments efficiently. In *WWW*, pages 61–70. ACM, 2009.

[14] A. Hampapur and R. M. Bolle. Comparison of distance measures for video copy detection. In *ICME*. IEEE Computer Society, 2001.

[15] A. Hampapur, K. Hyun, and R. M. Bolle. Comparison of sequence matching techniques for video copy detection. In *Storage and Retrieval for Media Databases*, volume 4676 of *SPIE Proceedings*, pages 194–201. SPIE, 2002.

[16] M. R. Henzinger. Finding near-duplicate web pages: a large-scale evaluation of algorithms. In *SIGIR*, pages 284–291. ACM, 2006.

[17] L. Jiang, G. Misherghi, Z. Su, and S. Glondu. Deckard: Scalable and accurate tree-based detection of code clones. In *ICSE*, pages 96–105. IEEE Computer Society, 2007.

[18] L.-W. Kang, C.-Y. Hsu, H.-W. Chen, and C.-S. Lu. Secure sift-based sparse representation for image copy detection and recognition. In *ICME*, pages 1248–1253. IEEE, 2010.

[19] C. Kim and B. Vasudev. Spatiotemporal sequence matching for efficient video copy detection. *IEEE Trans. Circuits Syst. Video Techn.*, 15(1):127–132, 2005.

[20] R. Koschke. Survey of research on software clones. In *Duplication, Redundancy, and Similarity in Software*, volume 06301 of *Dagstuhl Seminar Proceedings*. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), 2006.

[21] J. Law-To, L. Chen, A. Joly, I. Laptev, O. Buisson, V. Gouet-Brunet, N. Boujemaa, and F. Stentiford. Video copy detection: a comparative study. In *CIVR*, pages 371–378. 2007.

[22] C.-S. Lu and C.-Y. Hsu. Geometric distortion-resilient image hashing scheme and its applications on copy detection and authentication. *Multimedia Syst.*, 11(2):159–173, 2005.

[23] C.-S. Lu, C.-Y. Hsu, S.-W. Sun, and P.-C. Chang. Robust mesh-based hashing for copy detection and tracing of images. In *ICME*, pages 731–734. IEEE, 2004.

[24] H. A. Maurer, F. Kappe, and B. Zaka. Plagiarism - a survey. *J. UCS*, 12(8):1050–1084, 2006.

[25] A. Mittelbach, L. Lehmann, C. Rensing, and R. Steinmetz. Automatic detection of local reuse. In *EC-TEL*, volume 6383 of *Lecture Notes in Computer Science*, pages 229–244. Springer, 2010.

[26] K. Muthmann and A. Löser. Detecting near-duplicate relations in user generated forum content. In *OTM Workshops*, volume 6428 of *Lecture Notes in Computer Science*, pages 698–707. Springer, 2010.

[27] Z. Nie and S. Kambhampati. A frequency-based approach for mining coverage statistics in data integration. In *ICDE*, pages 387–398. 2004.

[28] Z. Nie, S. Kambhampati, and U. Nambiar. Effectively mining and using coverage and overlap statistics for data integration. *IEEE Trans. Knowl. Data Eng.*, 17(5):638–651, 2005.

[29] C. K. Roy and J. R. Cordy. Near-miss function clones in open source software: an empirical study. *Journal of Software Maintenance*, 22(3):165–189, 2010.

[30] C. K. Roy, J. R. Cordy, and R. Koschke. Comparison and evaluation of code clone detection techniques and tools: A qualitative approach. *Sci. Comput. Program.*, 74(7):470–495, 2009.

[31] S. Schleimer, D. S. Wilkerson, and A. Aiken. Winnowing: Local algorithms for document fingerprinting. In *SIGMOD*. 2003.

[32] J. Seo and W. B. Croft. Local text reuse detection. In *SIGIR*, pages 571–578. ACM, 2008.