MapReduce Data Processing to Data-Mine the Blogosphere

Project Ideas

Hohyon Ryu
School of Information
University of Texas at Austin
hohyon@utexas.edu

Jae Hyeon Bae
Computer Science
University of Texas at Austin
metacret@gmail.com

ABSTRACT

This article is to present project ideas for Fall 2011 Data-Intensive Computing for Text Analysis class. Using MapReduce-based text processing for keyword extraction, possible applications are discussed. First, using implicit links, sparse links between blog articles that hampers applying information retrieval techniques can be overcome. Second, with Latent Dirichlet Allocation, we may generate a topic model based on extracted keywords. Finally, with the extracted keywords, we can visualize blogs and keywords that may help perceive how the blogosphere looks like in the big picture.

Categories and Subject Descriptors

H.4.m [**Information Systems Applications**]: Miscellaneous; H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia

General Terms

MapReduce, Blogosphere

Keywords

Keyword Extraction, Data-intensive Processing, Implicit links in Blogosphere

1. INTRODUCTION

Blog is a very important and interesting source of diverse information. People search blogs to find out how to cook, what to do, what to see, and to discover what others think. Since blogs are usually written in plain language by individuals enthusiastic for their interest, blogs may be a lot more understandable than news articles and cover a very wide personal range of topics. However, unlike web information retrieval, blog information retrieval is still very limited and faces many challenges. Some well-established Web information retrieval techniques do not perform well on blogs because of sparse links, multimedia contents and the short length of the contents [1]. In particular, PageRank [3], which made a revolutionary improvement by taking into account the link structure of HTML Web documents, does not perform well for blog posts as they lack strong link structure. Another problem for blog information retrieval is the length of the blog posts. They vary from one line to several pages and require smoothing or normalization to achieve effective information retrieval. In this study, a novel method for augmenting implicit citation links to the explicit HTML links and document expansion for blog posts will be introduced. Extraction of implicit citation links from blog posts will address the sparse link problem that hampers utilizing PageRank and other link-based retrieval algorithms for blog information retrieval. The methodology of extracting implicit citation will utilize Blog content extraction using Decruft[1] and noun chunk extraction based on Python NLTK[2]. By augmenting documents implicit links, the proposed method tries to improve blog information retrieval. I will also explore possible other applications to improve information presentation for the Blogosphere.

2. PROJECT IDEAS

2.1 Blog Information Retrieval Performance Improvement Using Implicit Links

The idea of using link structure for information retrieval has been recognized as the key to successful information retrieval ever since the emergence of Google and its core algorithm, PageRank [3]. Another trial to make use of links for information retrieval was proposed was HITS [4], and PageRank and HITS are now the minimum standard of the web information retrieval to assure the quality of the retrieved web pages. However, unfortunately, these powerful algorithms cannot be fully exploited for blog information retrieval due to lack of strong link structure in the blogosphere.

With extracted keyphrases using MapReduce and NLTK, we will provide more prior information for a MapReduce-based search engine Ivory[3], to improve blog search performance.

Data-Intensive Computing for Text Analysis Fall 2011, School of Information, University of Texas at Austin

[1] http://www.minvolai.com/blog/decruft-arc90s-readability-in-python/

[2] Natural Language Toolkit, http://www.nltk.org/

[3] http://www.umiacs.umd.edu/ jimmylin/ivory/docs/index.html

2.2 Meme clustering and topic analysis

Clustering is the most basic technique to retrieve meaningful information from the corpus. In this project, we will use K-means clustering as the starting point. Since we do not have any prior information of meme distribution, we will apply Canopy clustering [?] to choose initial seeds of clustering instead of random seeds selection of K-means clustering. There are two threshold parameters $T_1, T_2, T_1 > T_2$ at Canopy clustering. Because this algorithm is pretty fast, we can experiment with various combination of threshold parameters to get reasonable number of clusters. After K-means clustering with centroids generated by Canopy algorithm, we can start analyzing meme distribution in each cluster and find which memes are releated each other. Cosine distance will be used as the distance measure between two memes. Suppose that document P, Q are represented as two vectors $(P_0, P_1, ..., P_n)$ and $(Q_0, Q_1, ..., Q_n)$, cosine distance between two documents can be expressed mathematically as the following:

$$D(P,Q) = 1 - \frac{\sum(P_i \times Q_i)}{\sqrt{\sum(P_i)^2 \times \sum(Q_i)^2}}$$

For Canopy generation and K-means clustering, Apache Mahout [?] implementation will be used. Latent Dirichlet Allocation (LDA)[2] is a generative probabilistic model that can be utilized to detect topics in a document collection. LDA models the document as the random mixture over latent topics and topic can be represented by a distribution over words. Using LDA, we can get meme mixtures for each topic and topic assignments for memes in each document. Thus, armed with LDA, we can group memes with similar topics and analyze many things including topic distrubition over documents and relationshp among topic space. Yahoo! LDA [?] implementation will be used. LDA algorithm also needs several parameters, of which the most critical parameter, the number of topics should be specified. This number can be same as K set with above K-means clustering. But for finding optimal the number of topics, we will calculate perplexity of test collection with the various number of topics. Formally, for a test set of M documents, the perplexity is

$$perplexity(D_{test}) = exp\{-\frac{\sum_{d=1}^{M} log\ p(\mathbf{w}_d)}{\sum_{d=1}^{M} N_d}\}$$

2.3 Visualization for Users

There has been many attempts to visualize blogosphere [5, 6], however they could hardly help users for better navigation. Using the visualization techniques integrated into the navigation system, we would be able to benefit actual users to find the blog articles better and to navigate between related articles easily.

3. ACKNOWLEDGMENTS

Jae Hyeon Bae will join with me for the final project.

4. REFERENCES

[1] N. Agarwal, H. Liu, L. Tang, and P. S. Yu. Identifying the Influential Bloggers in a Community. *Sites The Journal Of 20Th Century Contemporary French Studies*, pages 207–217.

[2] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

[3] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, Apr. 1998.

[4] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, Sept. 1999.

[5] C. Tauro, M. A. Pérez-Quiñones, P. Isenhour, S. Ahuja, and A. Kavanaugh. VizBlog: Discovering Conversations in the Blogosphere. In *Technology demonstration at Directions and Implications of Advanced ComputingConference on Online Deliberation*, page 6. University of California, Berkeley, 2008.

[6] M. Uchida, N. Shibata, and S. Shirayama. Identification and visualization of emerging trends from Blogosphere. In *Proceedings of International Conference on Weblogs and Social Media*, pages 305–306, 2007.