Hierarchically clustered technical blogs

Godfrey Winster S.
Department of Information Science and Technology
College of Engineering, Guindy Campus
Anna University, Chennai, India
godfrey@live.in

Swamynathan S.
Department of Information Science and Technology
College of Engineering, Guindy Campus
Anna University, Chennai, India
swamyns@annauniv.edu

ABSTRACT

Social network captivate huge number of users for learning, advertising, entertaining etc. Blogging is one of the key roles in social environment. Blogs are available in plenty for entertainment, business and educating the blog readers in the World Wide Web. The number of blogs available for technical discussion is numerous and the same can be accessed by the learner for gaining more knowledge from the web. In the current scenario if the blog reader searches the web for a topic, huge number of blogs are retrieved. These blogs are collection of different relevant, irrelevant and non-English language blogs. Blog readers face tedious problem in reading the relevant and useful blogs. In this paper a novel idea is proposed to cluster the blogs according to the document similarity using Hierarchical Agglomerative Clustering. This clustering uses the pre-computed similarity value. This clustering can work to give the users an overview of the contents of a document collection and makes the searching process easier. The experimental result shows that the proposed work yields better results than the other clustering approaches.

Categories and Subject Descriptors

I.2.4. [**Computing Methodologies**]: Artificial Intelligence - Knowledge Representation Formalisms and Methods - Semantic networks

General Terms

Algorithms

Keywords

Clustering, blogging, ontology, web mining, semantic web

1. INTRODUCTION

A weblog or blog is a special webpage on which an individual author (a blogger) or a group of collaborating authors periodically publish article (entries or posts) [13]. In recent years, there is an explosive growth in the use of web-based technology for distance learning systems. Nowadays learning in the web is considered to be more interesting. One prominent media is the blog creation. In 2010, Technorati reported to have indexed over 133,000,000 blogs since 2002. The vast numbers of blogs indicate an available source of information which can be used for the purpose of information retrieval. In every 24 hours there are 900,000 new blogs that are created and one cannot expect a user to be able to read and analyze whether these blogs are relevant to them or not. Even if the user resorts to such a means of manual search and processing of blogs it is going to be time consuming as well as tedious and by the time a user decides whether that particular blog is relevant or not, the user would have already spent a fair amount of his time over it. Social media contains huge volume of social entity which is relevant and irrelevant to the user. Searching the relevant blogs for the user query is the tedious task in blogosphere. Since the blog search is a time consuming process, it is necessary to summarize or cluster the blogs so that it will be easy for the user to retrieve the relevant blogs. In this paper we present a novel idea to cluster the XML blogs using hierarchical agglomerative clustering. This clustering of blogs is based on the similarity matrix. The similarity matrix is calculated using the cosine similarity of two blogs.

The contribution of this paper can be summarized as follows.

- Blogs are collected from the web using crawler based on ontology terms.
- Collected blogs converted to common XML format.
- Blogs are retrieved form the blog repository for specific subject in ontology.
- Retrieved blog are clustered and hierarchically organized to the user.

The remainder of this paper is organized as follows: Section 2 discusses the related work. Section 3 presents the block diagram of blog clustering system. Section 4 provides experimental results. Section 5 summarizes the conclusion and future work.

2. RELATED WORK

Recently, there has been a dramatic proliferation in the number of blogs. The growth of Weblogs or blogs on the internet has been phenomenal. Originally an online writing tool that

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICACCI'12, August 3-5, 2012, Chennai, T Nadu, India.
Copyright 2012 ACM 978-1-4503-1196-0/12/08…$10.00.

helped its users keep track of their own online records, the blog quickly turned into a key part of online culture. The method provides an easy way for an average person to publish material of any topic he or she wishes to discuss in a web site. With a popular issue, a blog can attract tremendous attention and exert great influence on society[4].

A blog social network has emerged as a powerful and potentially services-valued form of computer-mediated communication (CMC). More and more interactions take place in the blogosphere, combining the benefits of the accessibility of the web, the ease-of use of interface and the incentive of blogging (i.e. share, recommend, comment. . .etc.). Blog becomes a viral marketing site based on peer-production and it is promoted yet induced by online person to person interactions. Moreover, there exists a large number of information in the blogosphere, including text-based blog entries (articles) and profile, pictures or figures and multimedia resources. This becomes problematic for users. How do they deal with information overload problems and how do they effectively retrieve information they consider important? This gives us an incentive to develop a blog recommender approach and design an information filtering mechanism [14].

Current clustering approaches can be divided into two major categories, namely concept mapping and embedded methods. Concept mapping methods simply replace each term in a document by its corresponding concepts extracted from ontology before applying the clustering algorithm. Embedded methods, on the other hand, integrate the ontological background knowledge directly into the clustering algorithm[9]. Clustering algorithms can be used to reconstruct a topical hierarchy among tags, and suggest that these approaches may be used to address some of the weaknesses in current tagging systems. Tags are useful for grouping articles into broad categories, but less effective in indicating the particular content of an article[3]. The mountain views are generated using a tomographic clustering algorithm on the blog social network. The Mountain View shows mountains of communities consisting of connected blogs. Peaks and valleys of the Mountain View depict representative blogs as community authorities and community connectors, respectively[2].

Graph-based representation and k-Medoids algorithm are applied to cluster blog based on sentiment word. Structural information in the blog search results, namely the word occurring in the title or snippet, the order of words and the distance of the adjacent two words are also used for clustering [11]. FGW- K means algorithm for clustering blog data automatically calculated the weights for different feature groups [6]. Links to social media resources are often shared when writing about the same topic. The links between individual blog articles can be used to support this clustering with another dimension of information[5]. Preprocessing plays vital role in clustering blogs based on its content [12]. Community based blog clustering and ranking is done to associate the blog with certain community. The method selects the blogs that have performed actions to the seed posts over some threshold and the post that have received actions to the seed posts over some threshold which expands the blog community [10]. Information sources are available in the blogosphere in the form of tags or labels. The embedded latent relations are known as Label Relation Graph which is in the tags or labels are extracted using bloggers' collective wisdom. The label relation graph is used to compute similarity between tags and perform clustering. This collective wisdom based approach for blog clustering, termed as WisColl, is compared with a representative SVD-based approach that does not use collective wisdom to discern the differences[8]. Concept analysis based method is used to cluster the blogs in the blogosphere. Initial content of each blog entry is extracted using designed program. Preprocessing the raw data and by creating formal context and concept lattice, the concept similarity and concept ranking are obtained[7].

The works on clustering uses various parameters for clustering which gives different clusters for different clustering algorithms. Hence we introduce a similarity matrix based clustering algorithm which clusters all possible relevant blogs.

3. ARCHITECTURE OF HIERARCHICALLY CLUSTERED TECHNICAL BLOGS

Blog clustering system contains the modules like blog collection, blog preprocessing, blog repository, blog clustering algorithm as shown in Figure. 1. Ontology is used to collect the blogs. The blogs collected to demonstrate our algorithm has different subjects like 'Processor', 'thread' etc. A subject in ontology is the input to the crawler which collects all the blogs relevant to the subject. Blogs are collected from various blog sites such as wordpress.com, blogs.technet.com and blogspot.com. The blogs collected are converted to XML format. These may also contain noisy blogs which is removed in the preprocessing stage. Some blog may contain very less sentence which is very difficult to understand. Informal corpus used to specify a word or sentence is available in blogs. These are removed in preprocessing stage. The statistics of blogs collected are shown in Table 1.

Table 1: Statistics of blogs collected

Sl.No	Action	Number of blogs
1.	Number of subjects in ontology	306
2.	Number of blogs collected	15070
3.	Number of blogs after preprocessing	8355
4.	Number of blogs removed in preprocessing	6715

Ontology with subject and relation is given to the blog clustering algorithm. Blogs relevant to the subject are collected from the blog repository and the cosine similarity is calculated for blog bi and bj. Similarity matrix is created for the NxN blogs. The blogs with highest similarity value is clustered together first and then the similarity value is combined using the complete linkage (minvalue). Combining and clustering the blogs are done until all the blogs are under a single cluster. Finally the clustered blogs is obtained in the form of a tree structure. This hierarchical tree makes the searching process easier. Table 2 shows the step by step procedure for clustering.

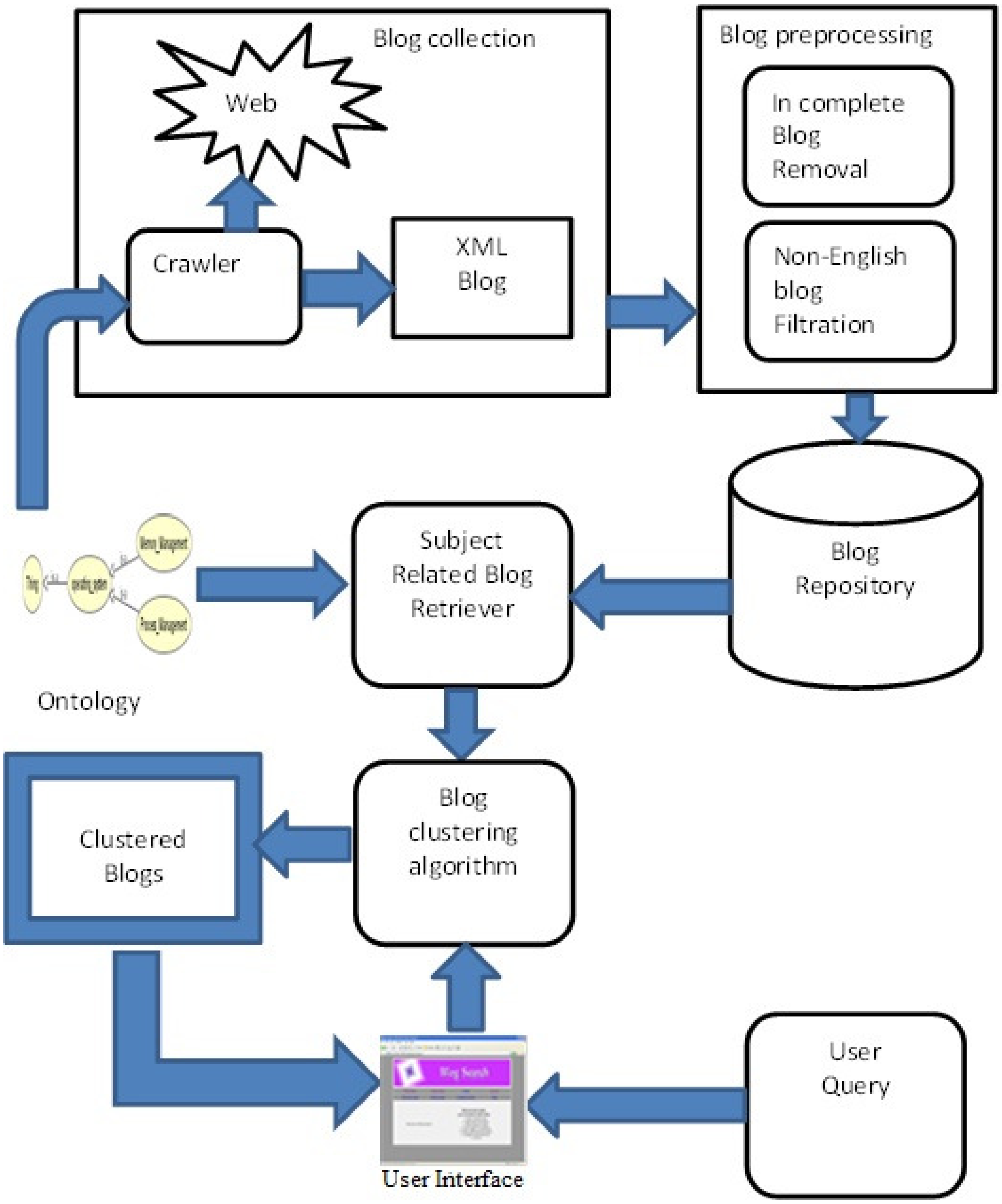


Figure 1: Architecture of Blog clustering system

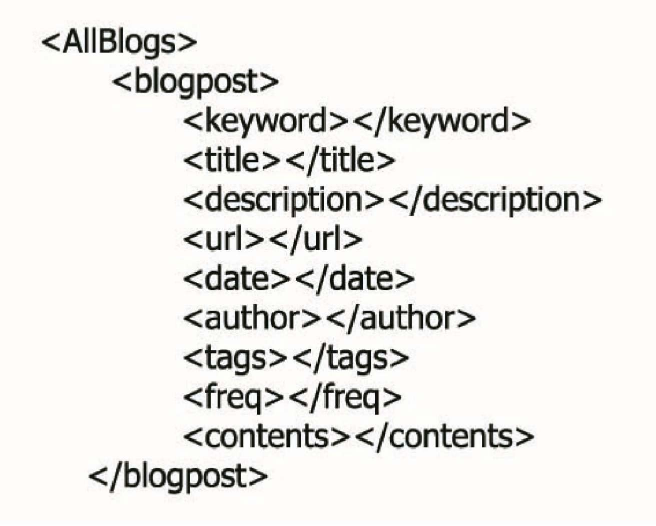
4. EXPERIMENTS AND RESULT ANALYSIS

Blogs are collected from the web using the crawler. Blogs collected are converted into the common XML format. The general XML format is shown in Figure 2 Collected blogs are in common XML format. Technical blogs are stored in the blog repository. Blogs which are not relevant and non-english blogs are removed at the preprocessing stage. Domain specific ontology is constructed to cluster the blogs in hierarchical order. Subject from the ontology is given to blog clustering algorithm which in turn create an agglomerative hierarchical cluster based on the similarity matrix.

Figure. 3 shows the initial matrix of 23 blogs collected from the blog repository for the subject ”thread”. Figure. 4 shows the final similarity matrix of the same set of blogs. Figure. 5 shows the final clustered blogs. Sample list of blogs with the rscore is shown in the Figure. 6 Rscore is calculated based on the similarity score. The blogs with highest similarity value is assigned highest rscore and the blog with less similarity value is assigned with less rscore. The URL list of all blogs clustered for a specific subject is listed in Table. 3 The blogs are also clustered using the Weka tool. The clustered set of blogs for different subject is shown in the Figure. 8 and Figure. 9. Figure. 8 shows the weka visualization of the blogs clustered for different subjects. Blogs clustered using the hierarchical algorithm in weka yields very less relevant blogs compared to our blog clustering algorithm. Figure. 9 shows the clustering of blogs based on the frequency of terms

Table 2: Agglomerative Blog Clustering using similarity matrix

Input : N Blogs (b1,b2......,bN), ontology subjects
Output : Clustered hierarchical tree
1.Let b1,b2,....,bN be the 'N' blogs retrieved for the subject from the blog repository
2.Initialize c1,c2,...cN cluster with one blog
3.Compute the cosine similarity score of 'N' blogs with one another
4.Create a similarity matrix for the NxN blogs
5.Cluster the blog with highest similarity value 6.Use complete linkage (min value) while combining the similar nodes.
7.Reduce the number of blogs by one(N–)
8.Repeat the step 3 to step 7 until all blogs are clustered as a single cluster



```
<AllBlogs>
    <blogpost>
        <keyword></keyword>
        <title></title>
        <description></description>
        <url></url>
        <date></date>
        <author></author>
        <tags></tags>
        <freq></freq>
        <contents></contents>
  </blogpost>
```

Figure 2: XML format of blog

which fails to cluster the blogs relevant to the concepts.

Cosine similarity is widely used in document clustering. Since we cluster blogs based on the similarity of blogs the cosine similarity is used. Similarity score is calculated based on the following formula [1].

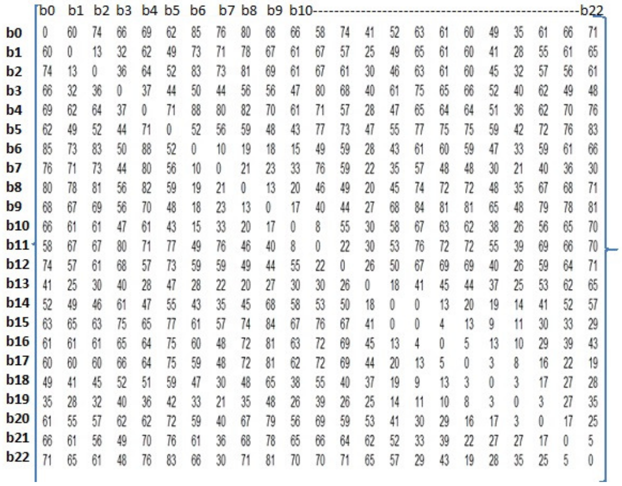
$$Cosine(b_j, b_q) = \frac{(\sum_{i=1}^{|v|} w_{ij} * w_{iq})}{(\sqrt{(\sum_{i=1}^{|v|} w_{ij}^2)} * \sqrt{(\sum_{i=1}^{|v|} w_{iq}^2)})} \qquad (1)$$

Where b_j, b_q are the blogs,($j \neq q$, j=1,2,....N, q=1,2,....N)

$w_{iq} = [0.5 + \frac{(0.5*f_{iq}}{(max f_{1q}, f_{2q}, ..., f_{|v|q}}] * log\frac{N}{bf_i}$

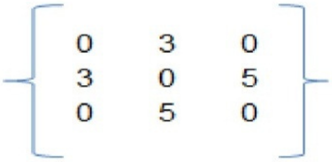
and $w_{ij} = tf_{ij} ibf_i$

Where$ibf_i = log\frac{N}{bf_i}$ and $tf_{ij} = \frac{f_{ij}}{(max[f_{1j}, f_{2j}, ..., f_{|v|j})}$



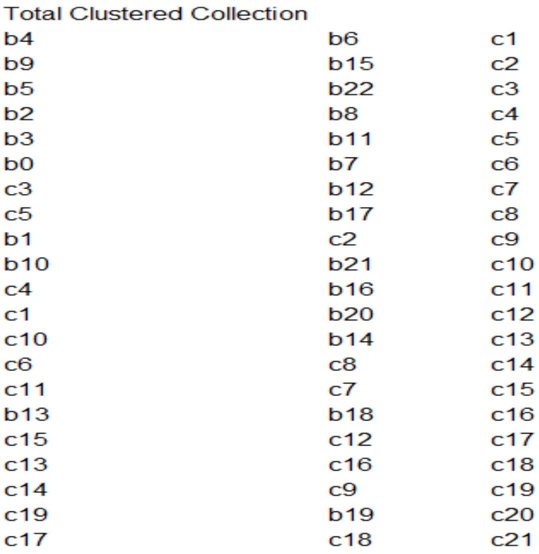
	b0	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10												b22
b0	0	60	74	66	69	62	85	76	80	68	66	58	74	41	52	63	61	60	49	35	61	66	71
b1	60	0	13	32	62	49	73	71	78	67	61	67	57	25	49	65	61	60	41	28	55	61	65
b2	74	13	0	36	64	52	83	73	81	69	61	67	61	30	46	63	61	60	45	32	57	56	61
b3	66	32	36	0	37	44	50	44	56	56	47	80	68	40	61	75	65	66	52	40	62	49	48
b4	69	62	64	37	0	71	88	80	82	70	61	71	57	28	47	65	64	64	51	36	62	70	76
b5	62	49	52	44	71	0	52	56	59	48	43	77	73	47	55	77	75	75	59	42	72	76	83
b6	85	73	83	50	88	52	0	10	19	18	15	49	59	28	43	61	60	59	47	33	59	61	66
b7	76	71	73	44	80	56	10	0	21	23	33	76	59	22	35	57	48	48	30	21	40	36	30
b8	80	78	81	56	82	59	19	21	0	13	20	46	49	20	45	74	72	72	48	35	67	68	71
b9	68	67	69	56	70	48	18	23	13	0	17	40	44	27	68	84	81	81	65	48	79	78	81
b10	66	61	61	47	61	43	15	33	20	17	0	8	55	30	58	67	63	62	38	26	56	65	70
b11	58	67	67	80	71	77	49	76	46	40	8	0	22	30	53	76	72	72	55	39	69	66	70
b12	74	57	61	68	57	73	59	59	49	44	55	22	0	26	50	67	69	69	40	26	59	64	71
b13	41	25	30	40	28	47	28	22	20	27	30	30	26	0	18	41	45	44	37	25	53	62	65
b14	52	49	46	61	47	55	43	35	45	68	58	53	50	18	0	0	13	20	19	14	41	52	57
b15	63	65	63	75	65	77	61	57	74	84	67	76	67	41	0	0	4	13	9	11	30	33	29
b16	61	61	61	65	64	75	60	48	72	81	63	72	69	45	13	4	0	5	13	10	29	39	43
b17	60	60	60	66	64	75	59	48	72	81	62	72	69	44	20	13	5	0	3	8	16	22	19
b18	49	41	45	52	51	59	47	30	48	65	38	55	40	37	19	9	13	3	0	3	17	27	28
b19	35	28	32	40	36	42	33	21	35	48	26	39	26	25	14	11	10	8	3	0	3	27	35
b20	61	55	57	62	62	72	59	40	67	79	56	69	59	53	41	30	29	16	17	3	0	17	25
b21	66	61	56	49	70	76	61	36	68	78	65	66	64	62	52	33	39	22	27	27	17	0	5
b22	71	65	61	48	76	83	66	30	71	81	70	70	71	65	57	29	43	19	28	35	25	5	0

Figure 3: Initial similarity matrix for a query word "Thread" in blog collection



$$\begin{bmatrix} 0 & 3 & 0 \\ 3 & 0 & 5 \\ 0 & 5 & 0 \end{bmatrix}$$

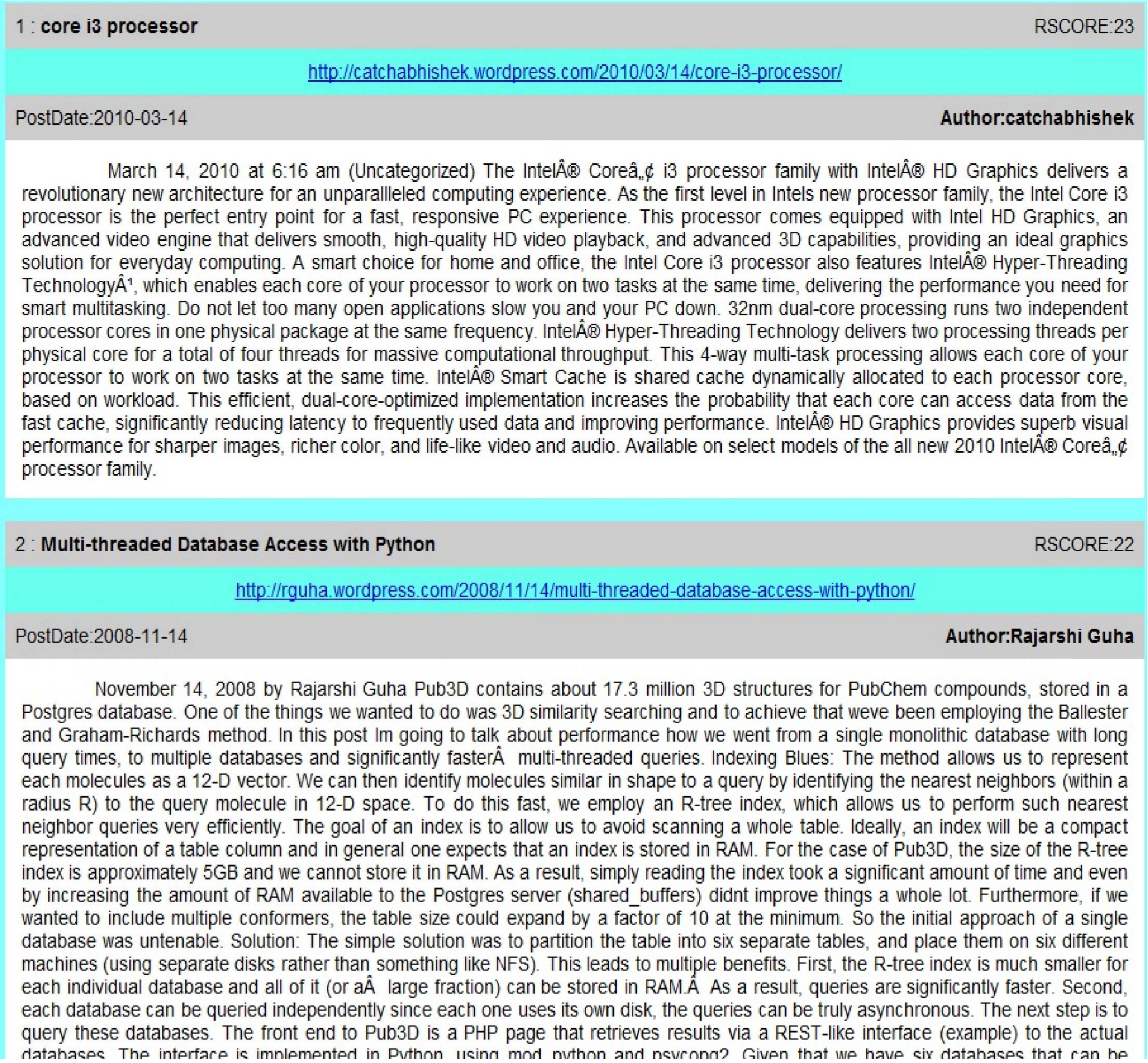
Figure 4: Final matrix for clustering



Total Clustered Collection		
b4	b6	c1
b9	b15	c2
b5	b22	c3
b2	b8	c4
b3	b11	c5
b0	b7	c6
c3	b12	c7
c5	b17	c8
b1	c2	c9
b10	b21	c10
c4	b16	c11
c1	b20	c12
c10	b14	c13
c6	c8	c14
c11	c7	c15
b13	b18	c16
c15	c12	c17
c13	c16	c18
c14	c9	c19
c19	b19	c20
c17	c18	c21

Figure 5: Clustered Blogs

Where N is the total number of blogs
bfi→number of blogs in which term ti appears at least once
fij→ frequency count of term ti in blog bj
|v|→vocabulary size of the blog collection



1 : **core i3 processor** RSCORE:23

http://catchabhishek.wordpress.com/2010/03/14/core-i3-processor/

PostDate:2010-03-14 **Author:catchabhishek**

March 14, 2010 at 6:16 am (Uncategorized) The IntelÂ® Coreâ„¢ i3 processor family with IntelÂ® HD Graphics delivers a revolutionary new architecture for an unparallleled computing experience. As the first level in Intels new processor family, the Intel Core i3 processor is the perfect entry point for a fast, responsive PC experience. This processor comes equipped with Intel HD Graphics, an advanced video engine that delivers smooth, high-quality HD video playback, and advanced 3D capabilities, providing an ideal graphics solution for everyday computing. A smart choice for home and office, the Intel Core i3 processor also features IntelÂ® Hyper-Threading TechnologyÂ¹, which enables each core of your processor to work on two tasks at the same time, delivering the performance you need for smart multitasking. Do not let too many open applications slow you and your PC down. 32nm dual-core processing runs two independent processor cores in one physical package at the same frequency. IntelÂ® Hyper-Threading Technology delivers two processing threads per physical core for a total of four threads for massive computational throughput. This 4-way multi-task processing allows each core of your processor to work on two tasks at the same time. IntelÂ® Smart Cache is shared cache dynamically allocated to each processor core, based on workload. This efficient, dual-core-optimized implementation increases the probability that each core can access data from the fast cache, significantly reducing latency to frequently used data and improving performance. IntelÂ® HD Graphics provides superb visual performance for sharper images, richer color, and life-like video and audio. Available on select models of the all new 2010 IntelÂ® Coreâ„¢ processor family.

2 : **Multi-threaded Database Access with Python** RSCORE:22

http://rguha.wordpress.com/2008/11/14/multi-threaded-database-access-with-python/

PostDate:2008-11-14 **Author:Rajarshi Guha**

November 14, 2008 by Rajarshi Guha Pub3D contains about 17.3 million 3D structures for PubChem compounds, stored in a Postgres database. One of the things we wanted to do was 3D similarity searching and to achieve that weve been employing the Ballester and Graham-Richards method. In this post Im going to talk about performance how we went from a single monolithic database with long query times, to multiple databases and significantly fasterÂ multi-threaded queries. Indexing Blues: The method allows us to represent each molecules as a 12-D vector. We can then identify molecules similar in shape to a query by identifying the nearest neighbors (within a radius R) to the query molecule in 12-D space. To do this fast, we employ an R-tree index, which allows us to perform such nearest neighbor queries very efficiently. The goal of an index is to allow us to avoid scanning a whole table. Ideally, an index will be a compact representation of a table column and in general one expects that an index is stored in RAM. For the case of Pub3D, the size of the R-tree index is approximately 5GB and we cannot store it in RAM. As a result, simply reading the index took a significant amount of time and even by increasing the amount of RAM available to the Postgres server (shared_buffers) didnt improve things a whole lot. Furthermore, if we wanted to include multiple conformers, the table size could expand by a factor of 10 at the minimum. So the initial approach of a single database was untenable. Solution: The simple solution was to partition the table into six separate tables, and place them on six different machines (using separate disks rather than something like NFS). This leads to multiple benefits. First, the R-tree index is much smaller for each individual database and all of it (or aÂ large fraction) can be stored in RAM.Â As a result, queries are significantly faster. Second, each database can be queried independently since each one uses its own disk, the queries can be truly asynchronous. The next step is to query these databases. The front end to Pub3D is a PHP page that retrieves results via a REST-like interface (example) to the actual databases. The interface is implemented in Python, using mod_python and psycopg2. Given that we have six databases that can be

Figure 6: Blog retrieved for the keyword ”thread” from the cluster

Similarity matrix of blogs are computed using the equation (1) and the initial matrix is shown in the Figure. 3 The highest similarity is between the blogs b4 and b6 and the same is clustered. To make the demonstration easier the computed value is multiplied by 10^2 and rounded to the integer value. The clustering algorithm continues till all the blogs are clustered as a single cluster. The final matrix for last round of clustering is shown in Figure. 4. The blogs after clustering is shown as a tree format in Figure. 7

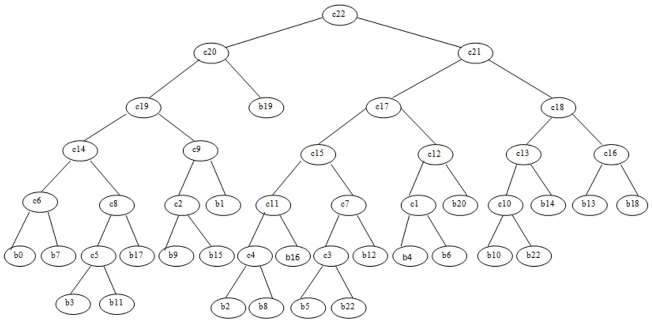
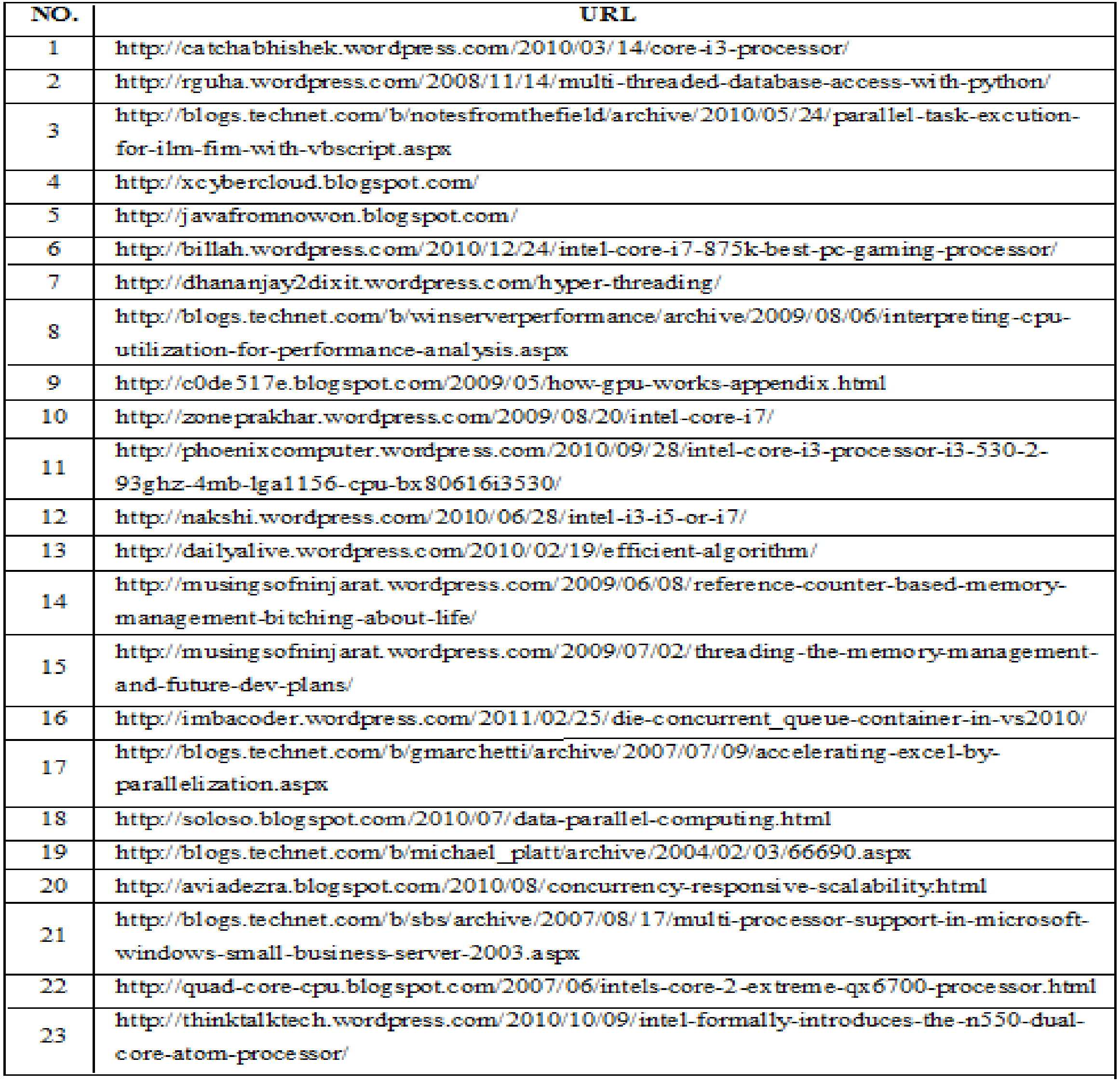


Figure 7: Tree structure of the clustered blogs for subject ”Thread”

5. CONCLUSION AND FUTURE WORK

The blogs are collected from the web using ontology and the blogs are clustered using the cosine similarity. The similarity matrix is constructed to cluster the blog in hierarchical order which yields very good result and list the blogs to the user. Search time is reduced while searching the clustered tree. The same can be clustered using Weka and the result of

Table 3: URL of all Blogs listed in descending order of the rscore



NO.	URL
1	http://catchabhishek.wordpress.com/2010/03/14/core-i3-processor/
2	http://rguha.wordpress.com/2008/11/14/multi-threaded-database-access-with-python/
3	http://blogs.technet.com/b/notesfromthefield/archive/2010/05/24/parallel-task-excution-for-ilm-fim-with-vbscript.aspx
4	http://xcybercloud.blogspot.com/
5	http://javafromnowon.blogspot.com/
6	http://billah.wordpress.com/2010/12/24/intel-core-i7-875k-best-pc-gaming-processor/
7	http://dhananjay2dixit.wordpress.com/hyper-threading/
8	http://blogs.technet.com/b/winserverperformance/archive/2009/08/06/interpreting-cpu-utilization-for-performance-analysis.aspx
9	http://c0de517e.blogspot.com/2009/05/how-gpu-works-appendix.html
10	http://zoneprakhar.wordpress.com/2009/08/20/intel-core-i7/
11	http://phoenixcomputer.wordpress.com/2010/09/28/intel-core-i3-processor-i3-530-2-93ghz-4mb-lga1156-cpu-bx80616i3530/
12	http://nakshi.wordpress.com/2010/06/28/intel-i3-i5-or-i7/
13	http://dailyalive.wordpress.com/2010/02/19/efficient-algorithm/
14	http://musingsofninjarat.wordpress.com/2009/06/08/reference-counter-based-memory-management-bitching-about-life/
15	http://musingsofninjarat.wordpress.com/2009/07/02/threading-the-memory-management-and-future-dev-plans/
16	http://imbacoder.wordpress.com/2011/02/25/die-concurrent_queue-container-in-vs2010/
17	http://blogs.technet.com/b/gmarchetti/archive/2007/07/09/accelerating-excel-by-parallelization.aspx
18	http://soloso.blogspot.com/2010/07/data-parallel-computing.html
19	http://blogs.technet.com/b/michael_platt/archive/2004/02/03/66690.aspx
20	http://aviadezra.blogspot.com/2010/08/concurrency-responsive-scalability.html
21	http://blogs.technet.com/b/sbs/archive/2007/08/17/multi-processor-support-in-microsoft-windows-small-business-server-2003.aspx
22	http://quad-core-cpu.blogspot.com/2007/06/intels-core-2-extreme-qx6700-processor.html
23	http://thinktalktech.wordpress.com/2010/10/09/intel-formally-introduces-the-n550-dual-core-atom-processor/

our clustering algorithm is compared with weka clustering. Weka contains more outlier which is eliminated in our blog clustering algorithm. The experimental results show that our proposed work produces better clustering and makes the user search process easier. This work can also be extended to cluster the blogs with query based clustering which will cluster the entire collection when a query is posted.

6. REFERENCES

[1] B.Liu, *"Web Data Mining"*, Springer, 2003.

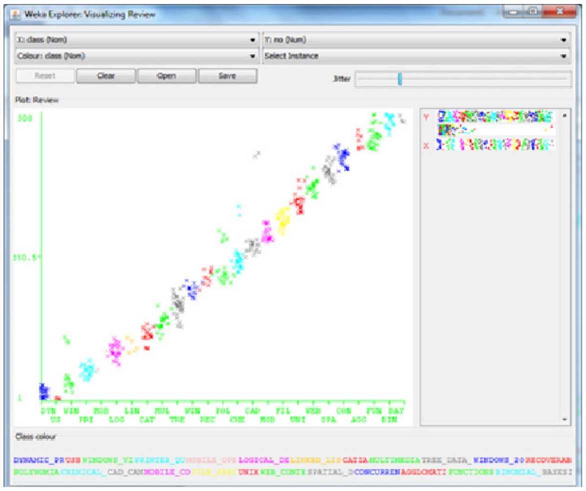
[2] B. L. Tseng, Junichi Tatemura, and Yi Wu, *"Tomographic clustering to visualize blog communities as mountain views"*, The 14th International World Wide Web Conference, Chiba, Japan, 2005, http://citeseerx.ist.psu.edu/viewdoc/summary?, doi=10.1.1.91.3409.

[3] Brooks, C.H., Montanez, N., *"Improved annotation of the blogosphere via autotagging and hierarchical clustering"*, Proceedings of the 15th International Conference on World Wide Web,2006, pp. 625-632, http://dl.acm.org/citation.cfm?doid=1135777.1135869.

[4] Chin-Lung Hsu, Judy Chuan-Chuan Lin, *"Acceptance of blog usage: The roles of technology acceptance social influence and knowledge sharing motivation"*, Journal of Information and Management, Vol. 45, 2008, pp. 56-74.

[5] Darko Obradovic, Fernanda Pimenta, Andreas Dengel, *"Mining Shared Social Media Links to Support Clustering of Blog Articles"*, International Conference on Computational Aspects of Social Networks (CASoN)), 19-21 October 2011, pp. 181-184, 10.1109/CASON.2011.6085940.

[6] Hongbo Li, Yunming Ye, Joshua Zhexue Huang, *"Improved blog clustering through automated weighting of text blocks"*, Proceedings of the Eighth International


Weka Explorer: Visualizing Review

X: class (Nom)

Y: no (Num)

Colour: class (Nom)

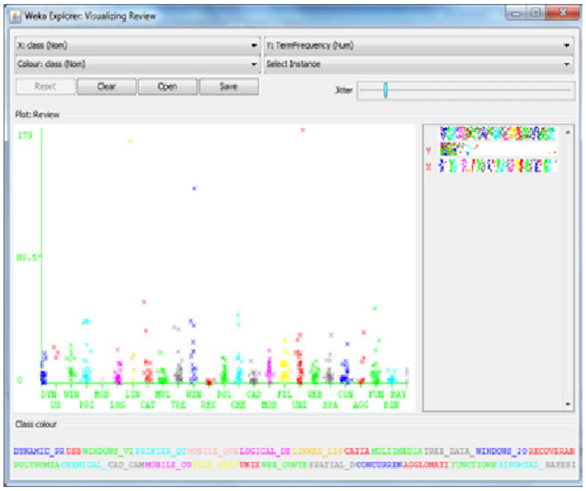
Select Instance

Jitter

Plot: Review

Class colour

Figure 8: Blogs clustered using weka


Weka Explorer: Visualizing Review

X: class (Nom)

Y: TermFrequency (Num)

Colour: class (Nom)

Select Instance

Jitter

Plot: Review

Class colour

Figure 9: Blogs clustered using weka using term frequency

Table 4: Blogs clustered in each Cluster

Clusters	Blogs
c6	b0, b7
c4	b2, b8
c5	b3, b11
c1	b4, b6
c3	b5, b22
c2	b9, b15
c16	b13, b18
c11	b2, b8 ,b16
c8	b3, b11, b7
c12	b4, b6, b20
c7	b5, b22, b12
c9	b9, b15, b1
c13	b10, b22, b14
c14	b0, b7, b3, b11, b17
c18	b10, b22, b14, b13, b18
c15	b2, b8, b16, b5, b22, b12
c19	b0, b7, b3, b11, b17, b9 ,b15, b1
c17	b2, b8, b16, b5, b22, b12, b4, b6, b20
c20	b0, b7, b3, b11, b17, b9, b15, b1, b19
c21	b2, b8, b16, b5, b22, b12, b4, b6,b20, b10, b22, b14, b13, b18
c22	b0,b7,b3,b11,b17,b9,b15,b1,b19,b2,b8,b16, b5,b22,b12,b4,b6,b20,b10,b22,b14,b13,b18

Conference on Machine Learning and Cybernetics, Baoding, 12-15 July 2009, pp. 1586-1591, http://10.0.4.85/ICMLC.2009.5212352.

[7] Jing Gao and Wei Lai, *"Formal Concept Analysis Based Clustering for Blog Network Visualization"*, ADMA 2010, Part I, LNCS 6440, pp. 394-404,http://dl.acm.org/citation.cfm?id=1947639.

[8] Nitin Agarwal, Magdiel Galan, Huan Liu, Shankar Subramanya, *"WisColl: Collective wisdom based blog clustering"*, Journal Information Sciences,2010 , Vol. 180 , pp. 39-61.

[9] Samah Fodeh, Bill Punch, Pang-Ning Tan, *"On ontology document clustering using core semantic features"*, Journal of Knowledge Information system,2011, Vol. 28, pp. 395 Ű 421.

[10] Seok-Ho Yoon, Jung-Hwan Shin, Sang-Wook Kim, Sunju Park, Jae Bum Lee, *"Subject-based extraction of a latent blog community"*, Journal of Information Sciences,2012, Vol. 184, pp. 215 Ű 229.

[11] Shi Fenga, Jun Pang, Daling Wang, Ge Yu, Feng Yang, Dongping Xu, *"A novel approach for clustering sentiments in Chinese blogs based on graph similarity"*, Journal of Computers and Mathematics with Applications, 2011, Vol. 62, pp. 2770 Ű 2778.

[12] Tomas Kuzar, Pavol Navrat, *"Preprocessing of Slovak Blog Articles for Clustering"*, IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology 31 August 2010-3rd September 2010,Toronto, Canada, Vol. 1, pp.314-317.

[13] Yun Chi, Belle L. Tseng, Junichi Tatemura, *"Eigen-Trend: Trend analysis in the Blogosphere based on singular value decompositions"*, International conference on Information and knowledge management CIKMŠ06,2006, pp. 391-397, 10.1109/WI-IAT.2010.273.

[14] Yung-Ming Li , Ching-Wen Chen, *"A synthetical approach for blog recommendation: Combining trust, social relation and semantic analysis"*, Journal of Expert Systems with Applications, 2009, Vol.36 pp. 6536Ű6547.